(2)

DESIGNING HELP SYSTEMS USING A GOMS MODEL: PART 1 - AN INFORMATION RETRIEVAL EVALUATION

Jay Elkerton
Susan Palmiter

Center for Ergonomics
The University of Michigan

Technical Report C4E-ONR-3
July 21, 1989

ABSTRACT

Using the GOMS model (Card, Moran, and Newell, 1983), a help system was developed which was complete and well structured. The content of this help system was determined from the goals, operators, methods, and selection rules, needed to perform HyperCard™ authoring tasks. The index to these help methods, which was an integrated part of the system, was determined from the hierarchical goal tree provided by the GOMS analysis. To determine the effectiveness of using GOMS as a design aid for help systems, the GOMS help system was compared to a state-of-the-art help interface developed by Apple® Computer which was modified slightly for experimental purposes (Original help system). Two groups of 14 users, assigned to one of the two help systems, retrieved help information for 56 tasks separated into 4 sessions. The results indicated that the GOMS users were significantly faster than the Original users with the largest speed difference occurring in the first session. Moreover, user performance with GOMS help was relatively stable, while users of the Original help improved significantly over the sessions. Interestingly, users subjectively rated the GOMS help system higher than the Original help system. The experiment suggests that the GOMS users were provided an explicit structure in the help system which allowed them to easily find help information. This was quite unlike the Original users who had to learn where help information was located. Overall, the results from this information retrieval study indicate that a GOMS model can aid in the development of help systems which are easy to use, easy to learn, and well liked.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): Tech. Rep. C4E - ONR - 3	5 MONITORING ORGANIZATION REPORT NUMBER(S): Tech. Rep. C4E - ONR - 3
6a NAME OF PERFORMING ORGANIZATION: University of Michigan; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION: Office of Naval Research
6c ADDRESS (City, State, and ZIP Code): Center for Ergonomics, 1205 Beal Ave. - IOE Bldg., Ann Arbor, MI 48109-2117	7b ADDRESS (City, State, and ZIP Code): 800 N. Quincy St., Arlington, VA 22217-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION: Office of Naval Research; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: Contract: N 000 14-87-k-0740
8c ADDRESS (City, State, and ZIP Code): 800 N. Quincy St., Arlington, VA 22217-5000	10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
61153N 42	RR 04209	RR 042091	4429008

11 TITLE (Include Security Classification): (U) Designing help systems using a GOMS model: Part 1 - An information retrieval evaluation

12 PERSONAL AUTHOR(S): Jay Elkerton and Susan Palmiter

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical	FROM 88-8-15 TO 89-8-14	89-7-21	58

16 SUPPLEMENTARY NOTATION

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): >human-computer interaction, online help, information retrieval, ~~GOMS~~

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Using the GOMS model (Card, Moran, and Newell, 1983), a help system was developed which was complete and well structured. The content of this help system was determined from the goals, operators, methods, and selection rules, needed to perform HyperCard(TM) authoring tasks. The index to these help methods, which was an integrated part of the system, was determined from the hierarchical goal tree provided by the GOMS analysis. To determine the effectiveness of using GOMS as a design aid for help systems, the GOMS help system was compared to a state-of-the-art help interface developed by Apple(R) Computer which was modified slightly for experimental purposes (Original help system). Two groups of 14 users, assigned to one of the two help systems, retrieved help information for 56 tasks separated into 4 sessions. The results indicated that the GOMS users were significantly faster than the Original users with the largest speed difference occuring in the first session. Moreover, user performance with GOMS help was relatively stable, while users of the Original help improved significantly over the sessions. Interestingly, users subjectively rated the GOMS

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL: John J. O'Hare	22b TELEPHONE (Include Area Code): (202) 696-4502; 22c OFFICE SYMBOL: Code 1142PS

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

help system higher than the Original help system. The experiment suggests that the GOMS users were provided an explicit structure in the help system which allowed them to easily find help information. This was quite unlike the Original users who had to learn where help information was located. Overall, the results from this information retrieval study indicate that a GOMS model can aid in the development of help systems which are easy to use, easy to learn, and well liked.

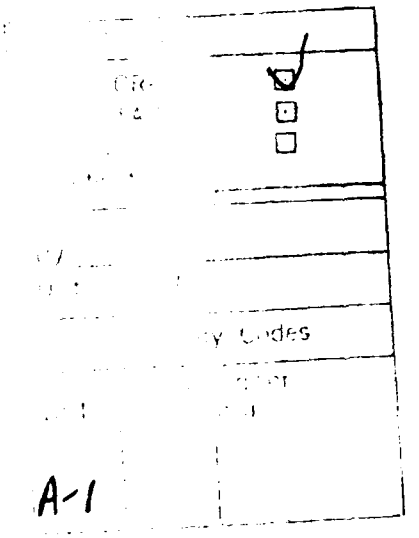


Table of Contents

Tables

Figures

INTRODUCTION

Do help systems actually help the user? If the goal of an online help system is to assist the user in accomplishing their computer-based tasks, the answer at this time is uncertain (Shneiderman, 1986). In reviewing online help research Shneiderman found that online help may be inferior to hard-copy manuals (Cohill and Williges, 1985; Dunsmore, 1980; Relles, 1979). However, in some of these studies there are data which suggest that online help is superior to no help (Cohill and Williges, 1985; Relles, 1979) and that online help can be improved (Borenstein, 1985; Magers, 1983). Yet, as indicated by Elkerton (1988) the specific aspects which improve the effectiveness of online help are relatively unknown.

This is disappointing. After all, online help is a safety net for users who may be having problems accomplishing their computer-based task. To build a better safety net there are three broad areas of knowledge which must be understood. Two areas are the content and presentation of help (see Wright, 1988), while the third is concerned with help access mechanisms (see Borenstein, 1985). Of the three, knowledge about help content is relatively undeveloped whereas research on help presentation and access is more widely available (see Elkerton, 1988; Houghton, 1984; Kearsley, 1988; Shneiderman, 1986). This may explain the equivocal research results when it is realized that help content must be addressed before decisions on help presentation and access can be made. For example, the research by Borenstein (1985) focused on help access methods, but revealed that differences in the quality of the help text may account for much of the improvement in the usability of the help interfaces.

Thus, this research focuses on how the content of an effective online help system can be specified based on theoretical models of human cognition. The model proposed for specifying the content of an online help system is the GOMS model (Card, Moran, and Newell, 1983) of human-computer interaction. Using this model, users can be provided information on **Goals** or meaningful tasks that can be accomplished, step-by-step **Methods** for accomplishing these goals, **Operators** or actions required of users, and if multiple interface methods exist, **Selection rules**, for choosing a specific method. The GOMS model provides the user with the

procedural knowledge required for operating the interface. This procedural knowledge is to be contrasted with the typical help information which consists of command lists containing the syntactic details for command use. This information is at the level of operators since users are provided the low-level details for interacting with the interface. Although operator-level knowledge may be useful for more experienced users, research suggests that novice users need to know what methods or procedures are necessary and available to accomplish their tasks (Elkerton, 1988; Wright, 1988).

Another reason for selecting GOMS as the model for help system content is that it is a widely documented model for human-computer interface design. Thus, this model could be used to develop a help system during the initial design stages rather than after the interface is developed. Using the same GOMS model to design the interface and the help documentation could result in a substantial reduction in development time and effort. Designers familiar with writing GOMS models for interface design could use these models to construct the help system which then could be evaluated with the initial prototype interface. This use of GOMS models is quite different from other efforts which have used GOMS to predict the time to use and learn interfaces (see Kieras, 1988). Instead, the emphasis here is on the qualitative content of the model rather than its quantitative predictions.

Using a help system consists of at least two components: retrieving appropriate help methods and then executing these methods in an interface task. This paper only presents the results from an information retrieval investigation. Our hypotheses were that users of a help system developed using GOMS would be faster and more accurate in finding appropriate help information. We suspected that structuring the help system using GOMS would explicitly cue users where appropriate help information was located. Specifically, the hierarchical goal structure of a GOMS model would serve as a direct indexing mechanism to detailed interface methods. We also thought that developing a help system using a GOMS model would lead to a more complete description of interface methods and thereby improve the accuracy of users retrieving help information. Providing this explicit, task-oriented structure and a complete description

of the interface methods also has been suggested by Duffy, Mehlenbacher, and Palmer (in press) as a procedure for designing and evaluating online help systems.

To test our hypotheses we chose authoring tasks from HyperCard™ on the Apple® Macintosh™ computer as our learning environment. Choosing HyperCard allowed us to develop the experimental task environment quickly. More importantly, HyperCard is distributed with an online help stack (hereafter called the "Original" help stack) which we used as a state-of-the-art comparison system. This help stack is implemented in hypertext, emphasizes both procedural (how-to-use-it) and conceptual (how-it-works) knowledge, uses both textual and pictorial presentations, and uses a variety of help access techniques. Thus, the Original help stack has many features which may be beneficial to help system usability. Comparisons between it and a help stack created using GOMS (hereafter called the GOMS help stack) would indicate how important an explicit, procedural model is to help system usability.

METHOD

Participants

Thirty-four staff and students at the University of Michigan volunteered and received nominal payment for participating in this study. Of the 34 people, the data of 3 were excluded because of software failures, the data of 2 were removed because of time constraints, and 1 participant was excluded due to a failure to pass a criterion test on the HyperCard application. The remaining 14 males and 14 females were assigned equally to each experimental group so that the two groups had approximately the same level of experience using the Apple Macintosh.

All participants were required to have a minimum of 4 months of experience using the Macintosh with exposure both to a word processor (e.g., Microsoft® Word or MacWrite™) and a graphics application (e.g., MacDraw™ or MacPaint™). However, all participants were required to have never used HyperCard. All participants filled out a computer-based background questionnaire (developed using HyperCard!) regarding their computing experience. As shown Table 1, there was a wide range of experience in terms of use of the Macintosh, other

TABLE 1

Background Computing Experience of the Participants

Experience Measure			Median / Percentage
Macintosh Experience			
- Macintosh Use (Months of use)			25
- Frequency of Mac use (Times per week)			3
- MacWrite (Months of use)			12-24
- MacWrite (Times per week)			1-3
- MS Word (Months of use)			<4
- MS Word (Times per week)			<1
- MacPaint (Months of use)			4-12
- MacPaint (Times per week)			<1
- MacDraw (Months of use)			9-12
- MacDraw (Times per week)			<1
- CricketGraph (Months of use)			<4-8
- CricketGraph (Times per week)			<1
Computer Classes number of classes where computer applications were learned			2
Number of Programming Languages and non-Macintosh Machine Used			
- machines (8 given)			3
- languages (10 given)			3
How much programming have you done on the Mac? (1=never to 5=experienced)			1
Programming Experience (subjective rating assigned by the participant; 1 = beginner, 5 = expert)			2
			Percentage
Have you ever used the Toolkit on the Mac?	yes	-	11%
	no	-	89%
Do you know what an authoring language is?	yes	-	11%
	no	-	89%
Do you know any authoring languages?	yes	-	4%
	no	-	96%
Have you ever used any HyperText or Hypermedia systems?	yes	-	4%
	no	-	96%
Use of Computer			
- home			11%
- school/work			89%

computers, and a wide variety of computer languages. We thought this general experience with computers, specifically the Macintosh, and the lack of experience with HyperCard was representative of many new users of HyperCard. Moreover, it gave us an opportunity to study the use of a help system by a group of experienced Macintosh users who wanted to know how to use a new application.

Equipment and Software

The experiment was conducted on an Apple Macintosh II computer with an 11-inch monochrome display. The Macintosh was equipped with 1-megabyte of memory and was running HyperCard. The experimental software was written in HyperTalk™ and included software to log responses from users at a resolution of 10 responses/s. Although a keyboard was available during the experiment, participants rarely used it and primarily used the mouse to point and click on buttons.

HyperCard Help Stacks

GOMS help stack. To test whether a GOMS model provides better procedural structure and material for a help system required us to focus on whether the content made a difference rather than the presentation or access mechanisms for the help. Therefore, the presentation for the GOMS help stack was limited strictly to written text. In terms of access mechanisms, we used the same methods in the GOMS help stack as in the Original help stack. Consequently, we limited ourselves to the use of index tabs, arrow buttons, square buttons, and invisible word buttons marked with asterisks. Figure 1 shows these buttons and gives a brief explanation of their use. Examples of the help stacks in the following sections will present instances in which the access mechanisms were used.

The design of the GOMS help stack closely followed a GOMS analysis of the HyperCard authoring task. A full GOMS analysis revealed that 128 methods were required for the HyperCard authoring task and resulted in a GOMS help stack of 175 cards. Due to the size of this analysis, a full presentation of the GOMS model or the help stack cannot be made. For this

paper, a portion of the analysis and pieces of the GOMS help stack will be presented to give the reader an understanding of the help stack.

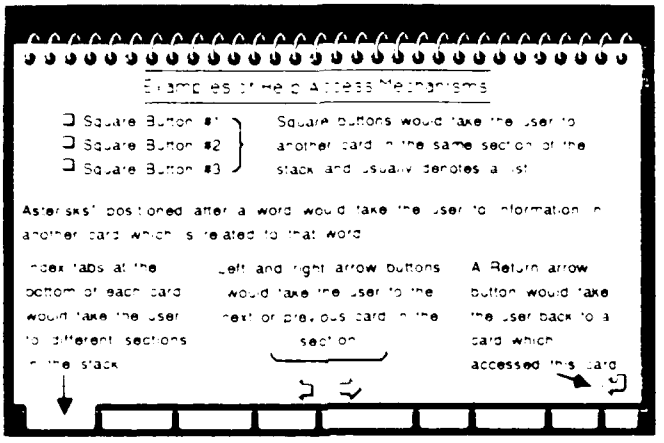


Figure 1. *Button access mechanisms used in the help stacks.*

As an example, consider the task of deleting a field. In the experiment, this task was phrased: "Get rid of a selected field." The relevant GOMS methods needed to execute this task appear in Table 2. The specific notation used in this description is "Natural GOMS Language" (NGOMSL) and was developed by Kieras (1988). NGOMSL was used because there is almost a direct translation to procedural directions once a method is described in this language. In fact, comparison of the NGOMSL for deleting a field in Table 2 with the procedural directions for that task in Figure 2 illustrates the close resemblance between the two descriptions.

As shown in Figure 2, upon entering the GOMS help stack users would see Card 1 which contains a list of the possible actions on index tab buttons. Users could then click on an index tab, such as "Delete," to get more information on the specific goals that can be accomplished. Going to Card 2 shown in Figure 2, actions and objects are combined to indicate that there are interface methods for deleting stacks, backgrounds, cards, buttons, fields, text in fields, and text in pictures. Users can then click on the appropriate square button, in this case "Fields," to go to another card with the detailed method.

TABLE 2

NGOMSL Methods Used to Delete a Field

Method to accomplish the goal of deleting the field
    Step 1: Decide: If necessary, Then accomplish the goal of selecting the field
    Step 2: Accomplish the goal of using a specific field delete method
    Step 3: Report goal accomplished

Method to accomplish goal of selecting the field
    Step 1: Decide: If necessary, Then use the Browse tool to go to the card with the field
    Step 2: Choose the Field tool in the Tools menu
    Step 3: Note that the fields on the card and background are displayed
    Step 4: Click on the field to be selected
    Step 5: Report goal accomplished

Selection rule set for goal of using a specific field delete method
    If you may want to paste the field somewhere else,
        Then Choose "Cut Field" from the Edit menu
    If you want to permanently delete the field,
        Then Choose "Clear Field" from the Edit menu
        (Alternative operator: Press delete on the keyboard)
    Report goal accomplished

Up to now, the only part of the NGOMSL model that has been discussed is an action-object table which maps HyperCard actions and objects to goals that can be accomplished. However, once users proceed to the detailed methods each statement in the NGOMSL description has a specific translation into the help directions. For example, in the NGOMSL description in Table 2 and on Card 3 in Figure 2, the first step of the field deletion method consists of deciding if the field needs to be selected and if necessary accomplishing that goal. The "Decide" operator in the NGOMSL model is used to pose a simple question to users. It is a judgement call on the part of the analyst (see Kieras, 1988) if users can determine whether a field is selected or not. The justification for this judgement call is that a selected field is readily apparent to users since it is outlined with a moving dashed line.

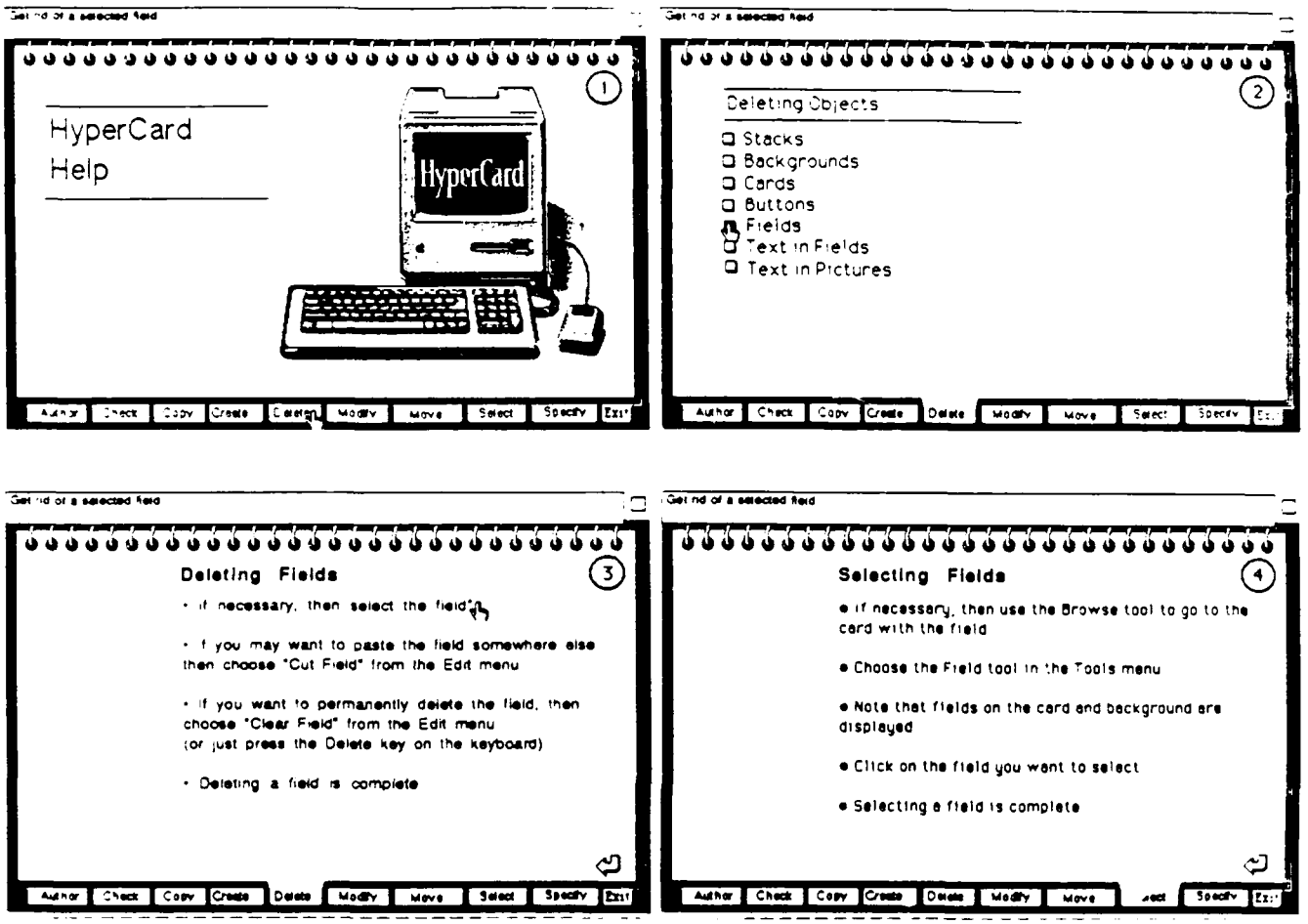


Figure 2. *GOMS help cards needed for determining how to delete a field.*

Goals are indicated in the method-level directions by words marked with asterisks. If users do not know how to select a field, then this button can be clicked to retrieve the detailed method. This selection is shown on the third card in Figure 2. In this way, users can retrieve as much procedural information as they need and exploit the adjustable level of detail that is inherent in the NGOMSL model.

As shown in Table 2 and on Card 3 in Figure 2, the next step indicates that users must make a decision on which deletion method should be executed. In encoding this selection rule, the analyst had to determine what task features are useful in deciding which method is most effective. Although specifying this information may seem arbitrary, leaving the decision up to

a novice user may be dangerous. In this case, a novice could accidently delete a field and never be able to retrieve it without knowing that "Clear" permanently deletes the field.

Another characteristic of the NGOMSL analysis, which is reflected in the help directions, is a judgement call on which operations users have knowledge about and need no further description (these are called high-level operators in NGOMSL). For example, in this NGOMSL analysis it was decided that users knew what the Browse tool was and could use it to go to other cards (see the field selection method in Table 2 and on Card 4 in Figure 2). In this experiment, moving with the Browse tool is a relatively rudimentary skill in HyperCard which users should know before entering the help stack since they were taught this in training. Therefore, there is no need for further analysis or to specify directions on browsing. If users required this information, the analysis and directions could be easily adapted by turning the high-level operator into a goal.

Two final characteristics of the use of NGOMSL for procedural directions need to be mentioned. First, a high-level operator was used to direct users' attention to the HyperCard display as shown in the third step of the field selection method in Table 2 (see also Card 4 in Figure 2). In this step, users would have to scan the display for the field that they wanted to delete. A judgement call was made not to analyze this operator further since there is sufficient information for users to follow and understand. The second feature of the NGOMSL model and the procedural directions is that users are told explicitly when they are finished executing a method (see the last steps of the methods in Table 2 and on Cards 3 and 4 in Figure 2). In this way, users can be sure that they do not need to perform any further actions and can proceed to accomplish other goals.

Original help stack. The Original help stack was modified for the experiment. Notably, material on HyperTalk was eliminated from this stack since the tasks would not require use of this programming language. In addition, an "Introduction" to HyperCard was eliminated from the Original help stack since it could trap users. In total, these modifications

reduced the Original help stack from 437 to 233 cards. Eliminating this material was believed to result in a fairer comparison with the GOMS help stack (and perhaps a more conservative one) since information not relevant to the authoring task was removed without unduly altering the structure of the Original help stack.

Differences between the GOMS and Original help stack can be seen by looking at the help cards a user might access when trying to find information on deleting a field. Six cards containing help information about how to delete a field are shown in Figure 3. Upon entering the Original help stack the user would see Card 1 that has a set of help topics listed on the index tabs. Perusal of these topics, however, reveals that they are not as action-oriented as the GOMS help stack. There are actions such as "Browse," "Paint," and "Copy," but the rest of the topics are general reference sections.

To retrieve information for deleting a field in the Original help stack, users would have to guess that the Reference section contains the relevant interface methods. They would then have to deduce that the topic "Creating/modifying fields," seen on Card 2 in Figure 3, has information on deleting fields. Clicking on this square button would take users to Card 3 which does not offer any information on deleting fields. However, if users were to click on the right arrow at the bottom of Card 3 they would see that Card 4 has an incomplete method for deleting a field. This method is incomplete since it fails to tell users how to select the field. To determine how to select a field, users would have to access Card A in Figure 3 which is contained in a different section (or one of several other cards with information on selecting a field).

The other way users could access the information contained on Card 4 in Figure 3 is to retrieve it through the index. This involves clicking on the "Index" topic tab and then moving through 8 cards until the topic "Field; delete, deleting" was found (see Card B in Figure 3). Clicking on this topic would then bring users to Card 4 directly.

This pattern of the help stack being organized differently, containing incomplete help methods, and having information spread over several cards throughout the stack held true for

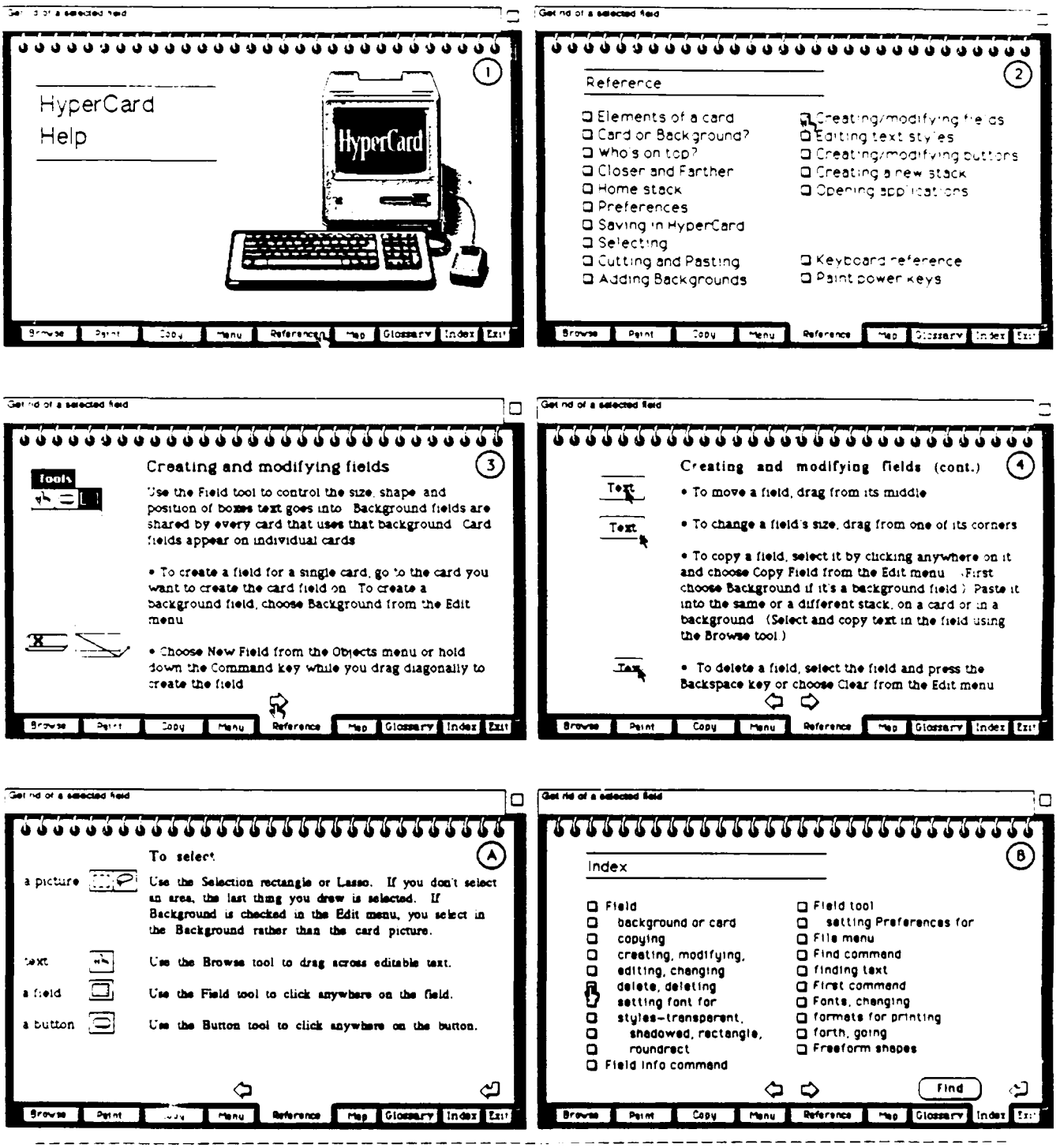


Figure 3. *Original help cards needed for determining how to delete a field.*

many tasks. It was suspected that users would have to navigate through several cards, perhaps not finding all of the information required for the task on any single card.

HyperCard Authoring Tasks

HyperCard authoring tasks involve manipulating (i.e., creating, copying, deleting, modifying, moving, selecting, specifying, and checking) objects (i.e., stacks, backgrounds, cards, buttons, fields, text in fields, and text in pictures). The 56 authoring tasks that users had to find in the help stacks are shown in Appendix A. The tasks were worded to avoid giving users any direct cues on where the help information was located in either help stack. In addition, the authoring tasks were chosen from an action-object analysis shown in Table 3. The

TABLE 3

Action-Object Analysis of the Authoring Tasks

	Objects						
Actions	Stack	Background	Card	Button	Field	*Text in* Fields	*Text in* Pictures
Check	2*	1	1	1		1	1
Create	1	1	1	1	1	1	1
Copy	1	1	1	1	2	1	1
Delete	1	no	1	2	1	1	2
Modify							
Move	nh	nh	1	2	2	1	1
Select	1	1	nh	1	1	1	2
Specify	1	1	1	4	3	1	1

* The numbers in each cell signify the number of tasks used in the experiment.
nh - Indicates that the task is not possible in HyperCard.
no - Indicates that the task is not documented in the original help stack.

goal of this analysis was to assure that a comprehensive set of authoring tasks was chosen for the experiment. As shown in Table 3, all actions were systematically sampled with the exception of modify. Modifying objects was not included since it combines other simpler tasks that were already sampled. Finally, one task, checking a field, was not chosen because interface methods were very similar to other tasks (i.e., checking a background, card, or button).

The multiple tasks in some of the cells of Table 3 allowed us to manipulate another characteristic of the authoring tasks. Specifically, the tasks were developed to require use of: (a) completely unique methods, (b) methods that share cards from previous tasks, (c) methods with a procedural structure similar to methods seen in previous authoring tasks, and (d) methods that are contained on exactly the same cards as previous authoring tasks. Examples of these four types of authoring tasks are shown in Table 4. The reader is referred to Appendix A for details on the specific relationships between tasks.

TABLE 4

Examples of the Four Types of Authoring Tasks

A. Unique Tasks
Task 10: Pick a button
Task 19: Relocate a selected button from one card to another card
B. Tasks Which Share Cards with Other Tasks
Task 4: Pick a stack
Task 15: Duplicate a selected background and put it into another stack
C. Tasks Which Have a Similar Procedural Structure with Other Tasks
Task 28: Duplicate a selected field and place it on the same card by using the option key
Task 42: Duplicate a selected button and place it on the same card by using the option key
D. Tasks Which Access the Same Cards as Other Tasks
Task 27: Get rid of a selected button
Task 55: Get rid of a selected button, but save a temporary duplicate of it

Unique tasks were defined as two tasks which share no help cards and have a unique procedural structure. Therefore, although Task 10 and Task 19 in Table 4 share a common object (buttons), they do not share cards since Task 19 indicates that the button is already selected. In terms of procedural structure the tasks are clearly different. Picking (selecting) a button requires users to click on it using the button tool. Whereas, relocating a selected button requires users to cut and paste it using menu commands.

Tasks which share cards with other tasks have a common card or cards which may be accessed when looking for help information. These tasks, however, do not have a similar procedural structure. This is particularly clear for the relationship between Tasks 4 and 15 in Table 4. Picking (selecting) a stack is part of the method for putting a background in another stack. Still, the tasks are fundamentally different in structure. Task 4 involves having users either click on a stack button or opening a stack with a menu command, while Task 15 involves having users cut and paste cards.

Tasks which have a similar procedural structure with other tasks are those tasks in which only the object of the interface method was changed. However, these tasks do not share any cards. For example, for Tasks 28 and 42 users would have to execute the same procedural steps with the exception that they would be duplicating a button or a field, respectively. This method similarity is pervasive in the GOMS help stacks due to the procedural consistency of the Apple Macintosh and HyperCard.

Finally, some tasks would access exactly the same cards as some other previous tasks. Therefore, GOMS users would have previously seen the help cards for Task 55 after having retrieving help information on how to get rid of a selected button (Task 27). We included these tasks to see if users could remember where help information was located.

The prediction was that the GOMS help stack would lead to faster and more accurate retrieval of the help methods which share or have similar help information (Task B, C, and D in Table 4). The GOMS help stack should be more memorable due to its strong relationship to the HyperCard tasks. This consistency with the tasks may have provided users a method to predict

where the appropriate help information was located. In contrast, since help information was not structured in this way in the Original help stack these effects would not be as pronounced.

Procedure

The experiment was conducted in two phases. The first phase was used for collecting background data, training users on HyperCard, and training them on experimental procedures. In total, this background and training phase lasted approximately 30 minutes. In the second phase of the experiment users looked for information on the 56 authoring tasks. This phase of the experiment lasted approximately 90 to 120 minutes.

Background and training. A paper and pencil background form was used to collect data on gender, age, field of study, year in school, and educational degrees. In addition, all users were asked how much overall experience they had using the Apple Macintosh (in months) and how frequently they used it during the week. Next, a background questionnaire was used to collect the data that appears in Table 1 of this report. This HyperCard questionnaire was useful for collecting this data quickly and automatically, and also familiarized users with moving through a stack by clicking on buttons.

Next, all users were given a short tutorial on how HyperCard works. This tutorial was also presented using HyperCard. The tutorial required users to browse through a set of cards and read about HyperCard concepts such as what are common HyperCard objects, what is the "home stack", how does one save in HyperCard, and what are the tools of HyperCard. This tutorial actively engaged users and often required them to practice what they just learned.

After users finished the tutorial on HyperCard concepts, they were given two tests: a true-false test on the concepts and a hands-on test on browsing skills. The true-false test on HyperCard concepts was presented using HyperCard and required users to answer all the questions on the test. If they had an incorrect answer on any of these items, they were informed and given additional information to read. After reviewing this material users were asked to correct their answers. This occurred repeatedly until all the answers were correct. Users

were monitored on how many times they needed to answer the questions. If users were incorrect 7 or more times, they were immediately excused from the experiment for failing to understand basic HyperCard concepts. Only one user was excused from the experiment for this reason.

Following the concepts test, users were asked to demonstrate their browsing skills. If users did not know how to perform a skill or did so incorrectly, the experimenter showed them how to do it correctly. Users were excused from the experiment if they failed to perform 3 or more of these browsing skills correctly. For this test, all users passed and were deemed capable of browsing through HyperCard stacks.

The final set of training tasks involved finding help information in a small HyperCard stack. Users were given three trials in which to find information. This practice gave users hands-on practice with the experimental procedures. Users were shown how to start a trial, indicate when they found the correct help information, how to end a trial, and how to use all the access mechanisms shown in Figure 1. Before starting the practice trials, all users were instructed to find all the essential information needed for executing each of the tasks and to do so as quickly and as accurately as possible.

Experimental trials. During the experimental trials users were instructed to locate help information for the 56 authoring tasks which were divided into 4 sets of 14 trials. The specific tasks assigned to each of the 4 sessions are given in Appendix A. This assignment was used to make sure that users had retrieved help information on simple tasks (e.g., selecting objects) before retrieving help information on more complex procedures. Within each session the task order was counterbalanced with a balanced Latin square.

To start a trial, users clicked on a button and then were presented a HyperCard display similar to Card 1 in Figures 2 and 3. Users were instructed to first look at the top of the display and to read the task fully before beginning their search. This task description was displayed during the entire trial. After understanding the task, users were to browse through the help stack looking for information which was essential for executing it. When users found

relevant information on a card, they were to "mark" the card by clicking on the check box in the top, right-hand corner of the display. Users were allowed to mark as many cards as they liked (but at least one card) and were also allowed to "unmark" cards.

When users felt they had found all the essential information, they simply clicked on the "Exit" index tab. Feedback was given as to whether they had found all the help information. This feedback was provided at one of three levels: (1) "Good! You found all the help information!!!," (2) "OK! You found some of the help information," and (3) "You did not find any of the help information." Feedback at level 1 was provided when users found all the possible information for that task in the specific help stack. This feedback was found to be essential for motivating users to perform the retrieval task accurately.

After completing each trial, users were presented a button to start the next trial. A break was given at the end of every session (14 trials). On all trials, a variety of data was automatically collected on how much time and which cards were marked. At the completion of all four sessions users were asked to fill out a subjective questionnaire, presented using HyperCard, on how they felt about the help stack and HyperCard in general. This HyperCard questionnaire appears in Appendix B. In addition, users also filled out an open-ended questionnaire about the experiment.

RESULTS

Retrieval Time and Effort

Help by session analyses. One indication of how much effort was required to find information is the number of cards browsed. A two-way analysis of variance (ANOVA) on the type of help stack (GOMS vs. Original) and the retrieval session (1 - 4) for the number of cards browsed revealed significant main effects for the type of help stack (F [1,26] = 7.89, p < 0.01) and retrieval session (F [3,78] = 17.46, p < 0.0001). In addition, a significant Help Stack by Retrieval Session interaction was found (F [3,78] = 28.80, p < 0.0001). As shown in Figure 4, there was a large difference between the help groups for the number of cards browsed

during the first retrieval session which diminished over subsequent sessions. *Post hoc* Newman-Keuls procedures (α = 0.05) indicated that there were significant differences between groups for every retrieval session except Session 4. Moreover, there were no differences between retrieval sessions for the GOMS help stack, while there were differences between the first session and the rest of the other sessions for the Original help stack. These results suggest that there were overall differences in the amount of material looked at between the two help groups (GOMS: 9.0 cards vs. Original: 12.7 cards) which was mediated by the retrieval session. Users of the GOMS stack looked at a constant amount of help information over the 4 retrieval sessions, whereas users of the Original stack decreased the amount of help information they browsed during the experiment.

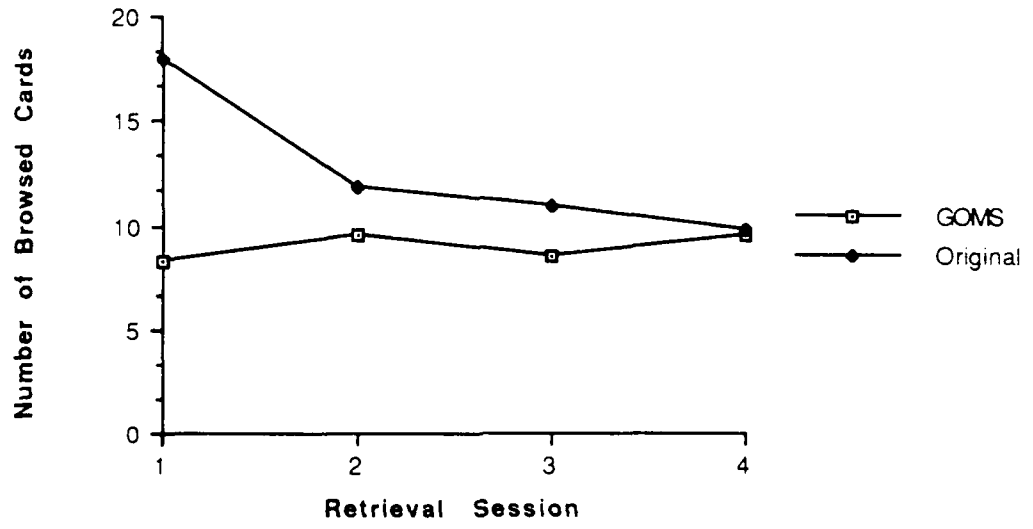


Figure 4. *Mean number of browsed cards for the help stacks across retrieval sessions.*

Not surprisingly, similar results were found for retrieval time. A two-way ANOVA on retrieval time revealed significant main effects for the type of help stack (F [1,26] = 8.41, p < 0.01) and retrieval session (F [3,78] = 62.67, p < 0.0001) with a significant Help Stack by

Retrieval Session interaction (F [3,78] = 39.70, p < 0.0001). As illustrated in Figure 5, the overall time difference between groups (GOMS: 69.1 s vs. Original: 89.9 s) was manifested largely in the first session. In this first session, users of the Original help stack required more than twice the amount of time to find help information. *Post hoc* Newman-Keuls procedures (α = 0.05) confirmed that there were significant differences between the help groups in Session 1 with no other significant differences between groups in the other sessions. In addition, there were few significant differences across retrieval sessions for GOMS users and a large significant difference between Session 1 and the rest of the other sessions for Original users.

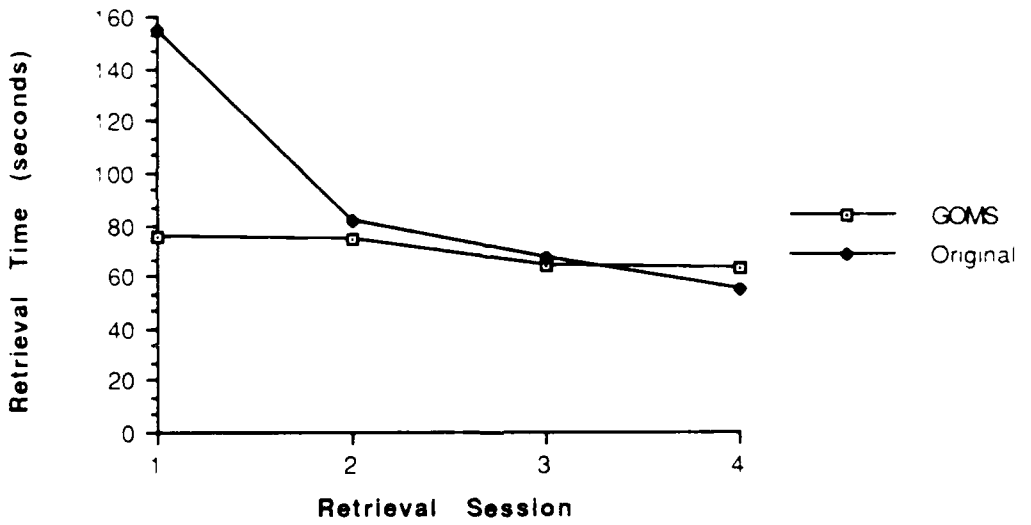


Figure 5. *Mean retrieval time for the help stacks across retrieval sessions.*

The number of cards browsed and retrieval time can be combined to form a measure of the number of cards browsed per minute to see how fast users looked through the cards. A two-way ANOVA on the number of cards browsed per minute revealed a marginally significant main effect for the type of help stack (F [1,26] = 3.27, p < 0.1), a highly significant effect for retrieval session (F [3,78] = 81.51, p < 0.0001), and a highly significant Help Stack by

Retrieval Session interaction (F [3,78] = 10.03, $p < 0.0001$). As shown in Figure 6, the two groups of users started out at approximately the same browsing rate. Then, each group increased its browsing rate in subsequent retrieval sessions. These increases in browsing rates were found to be significant using *post hoc* Newman-Keuls procedures ($\alpha = 0.05$) except between Sessions 2 and 3 for the GOMS users. In addition, there were significant differences in browsing rates between GOMS and Original users for every retrieval session except Session 1. This indicates that the Original users increased their browsing rate to a greater extent than the GOMS users.

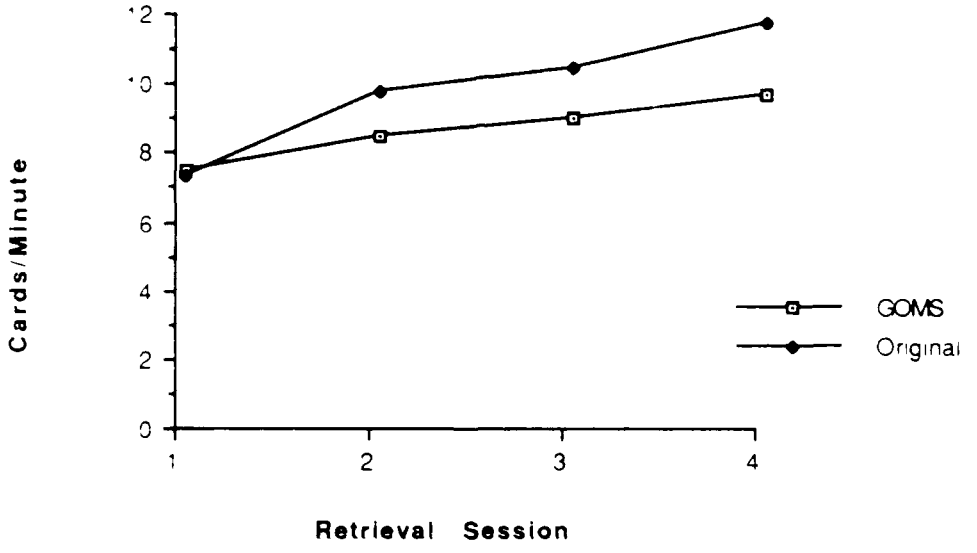


Figure 6. *Mean rate of browsing cards for the help stacks across retrieval sessions.*

To investigate the effort of finding the first piece of correct help information, the browsing time for users before they marked the first correct help card was analyzed. Only trials in which a correct card was marked were included in this analysis. An ANOVA on the time to mark the first correct card revealed significant main effects for the type of help stack (F [1,26] = 22.01, $p < 0.0001$) and retrieval session (F [3,78] = 32.08, $p < 0.0001$). Once

again, a significant interaction between type of help stack and retrieval session was observed (F [3,78] = 35.08, p < 0.0001). The data in Figure 7 show that Original users during the first session needed almost three times the amount of time to find and mark the first correct card than GOMS users. *Post hoc* Newman-Keuls procedures (α = 0.05) revealed that this was the only significant difference between help groups at each retrieval session. Moreover, the comparisons of means found that there were no differences between retrieval sessions for the GOMS users, while there was a significant difference between Session 1 and the rest of the other sessions for the Original help group. This result is important since users of the Original help stack could be tremendously frustrated spending close to 2 minutes before finding some relevant help information.

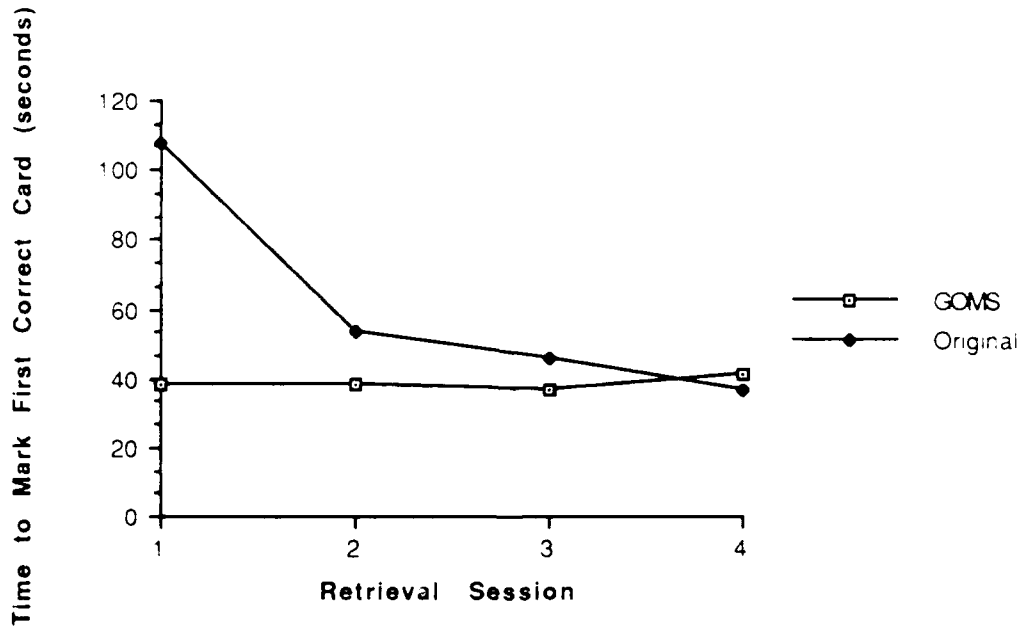


Figure 7. *Mean time to find the first piece of correct help information for the help stacks across retrieval sessions.*

Where were the users of the Original help stack spending this additional time in the first retrieval session? Were users wandering aimlessly through the help stack or were they trying

to decide if the help information was appropriate to the task? These questions can be partially addressed by looking at the type of help cards people accessed and how much time they spent on these cards. A way to classify help cards was whether they were navigational or informational. Navigational cards directed users to the content of the help methods. Examples of navigational cards for the GOMS help stack include Cards 1 and 2 in Figure 2 and for the Original help stack include Cards 1, 2, and B in Figure 3. In contrast, informational cards provided users detailed methods for executing authoring tasks. Examples of informational cards for the GOMS help stack include Cards 3 and 4 in Figure 2 and for the Original help stack include Cards 3, 4, and A in Figure 3.

An ANOVA on the number of navigational cards accessed during a trial revealed significant main effects for the type of help stack (F [1,26] = 15.60, p < 0.0005) and retrieval session (F [3,78] = 21.51, p < 0.0001) with the familiar interaction between type of help stack and retrieval session (F [3,78] = 14.58, p < 0.0001). The data for the interaction are shown in Figure 8. *Post hoc* Newman-Keuls procedures (α = 0.05) found that there was a larger number of navigational cards accessed by the Original users in the first retrieval session when compared to the rest of the other sessions. In addition, there were significant differences between help groups in each retrieval session and no significant differences between sessions for the GOMS users. These results imply that users of the Original help stack accessed a larger number of navigational cards throughout the experiment (Original: 6.6 cards vs. GOMS: 3.5 cards) while also needing to access a large number of these cards during the first retrieval session.

In terms of time on the navigational cards a similar result was found although the difference between the help groups over the retrieval sessions was not as consistent. An ANOVA found significant main effects for the type of help stack (F [1,26] = 40.13 , p < 0.0001) and retrieval session (F [3,78] = 100.42, p < 0.0001) with an interaction between type of help stack and retrieval session (F [3,78] = 50.47, p < 0.0001). The data for the interaction are shown in Figure 9, and illustrate that the largest significant difference was between the

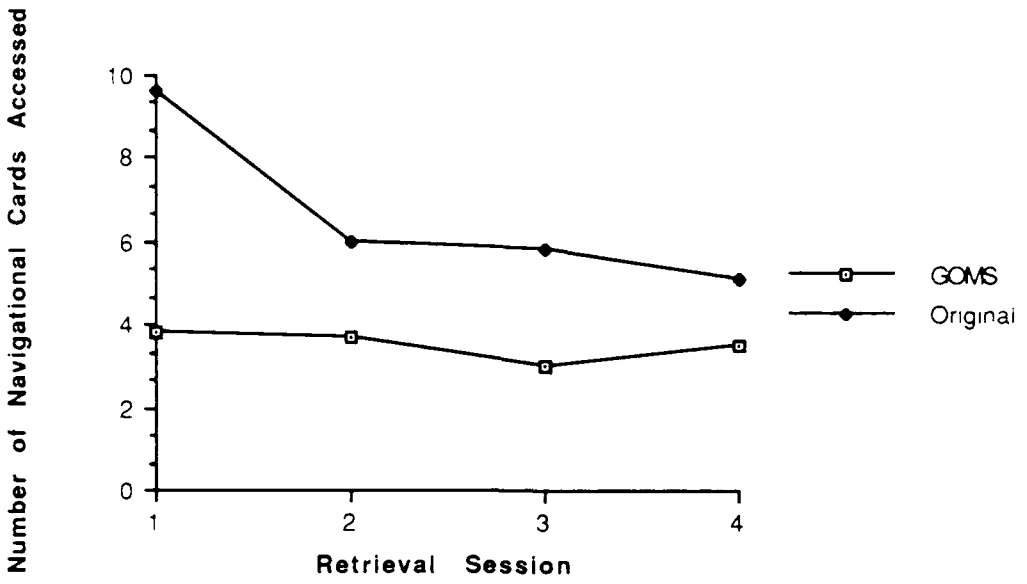


Figure 8. *Mean number of navigational cards accessed in the help stacks across retrieval sessions.*

Original users in Session 1 and the rest of the experimental conditions (Newman-Keuls, α = 0.05). These results indicate that users were taking more time to navigate through the Original help stack than the GOMS help stack during Session 1. Taking into consideration the total number of navigational cards browsed, Original users in subsequent retrieval sessions were going through the additional navigational cards faster than the rate at which GOMS users went through their navigational cards (see Figures 8 and 9).

For informational cards, very similar results were obtained for the number of cards accessed and the time spent on these cards. For this reason, only the data on time spent on informational cards will be reported. An ANOVA on this dependent variable found a significant main effect for retrieval session (F [3,78] = 30.31, p < 0.0001) and the interaction between type of help stack and retrieval session (F [3,78] = 20.87, p < 0.0001), but no significant main effect due to type of help stack (F [1,26] = 1.00, p > 0.3). The data for the interaction are shown in Figure 10. As illustrated the data are very similar to previous interactions with

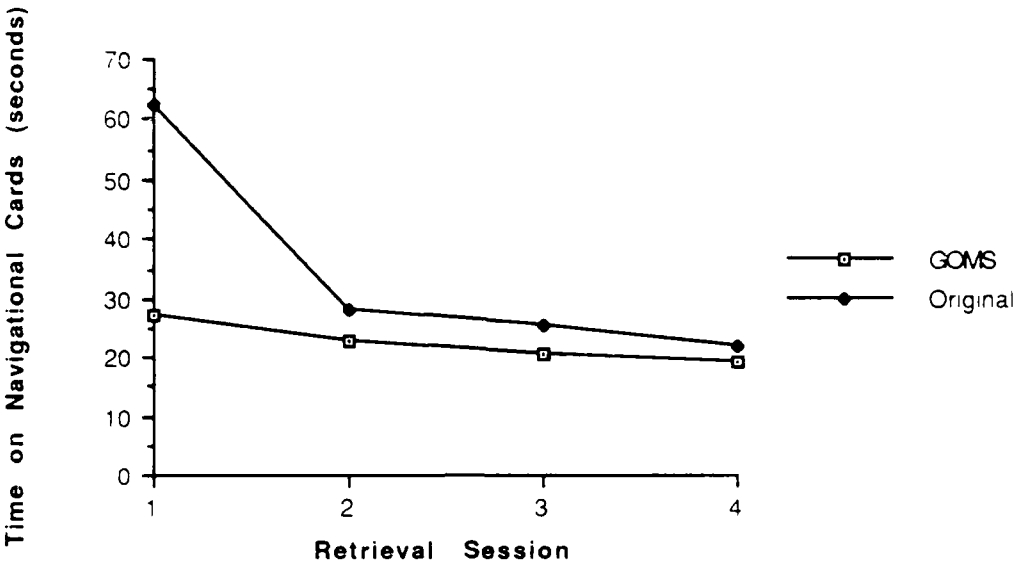


Figure 9. *Mean time on navigational cards in the help stacks across retrieval sessions.*

the major statistical difference residing between the first retrieval session of the Original help stack and the other experimental conditions (Newman-Keuls, $\alpha = 0.05$). Thus, users of the Original help stack accessed more informational cards and spent more time on them during the first retrieval session. However, the lack of any main effect for the type of help stack suggests that users of the two help stacks were accessing approximately the same number of informational cards and spending about the same amount of time on them overall.

Most of the analyses have shown that the difference between the help stacks was manifested largely in the first session. One hypothesis for these results may be that during the first session users of the Original help stack had to become familiar with the structure of the help stack and had to learn where help was located. Since the same group of authoring tasks was presented to all users during the first session, the results also may be due to the particular authoring tasks presented during the first session. However, a task-order explanation is less

plausible if one looks at the data which reflects the number of times users revisit help cards through the course of the experiment.

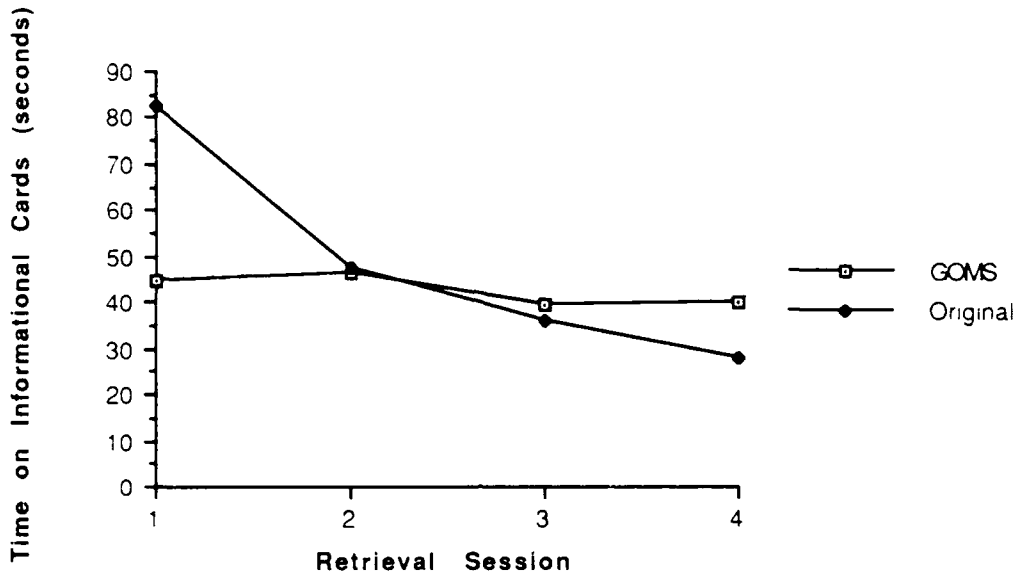


Figure 10. *Mean time on informational cards in the help stacks across retrieval sessions.*

The number of times users revisited cards that were accessed on previous trials was analyzed with an ANOVA and revealed significant main effects for the type of help stack (F [1,26] = 22.18, p < 0.0001) and retrieval session (F [3,78] = 19.81, p < 0.0001). In addition, a significant Help Stack by Retrieval Session interaction (F [3,78] = 9.86, p < 0.0001) was found. Beyond the slight and expected increase in the number of cards revisited shown in Figure 11, the results of *post hoc* Newman-Keuls procedures (α = 0.05) revealed that users of the Original help stack were revisiting help cards more frequently during all sessions. Therefore, with this more frequent revisiting one would suspect that users of the Original help stack would learn where information was located. In fact, based on the high-level of revisiting users of the Original stack seemed to be going to many of the same help cards for different authoring tasks.

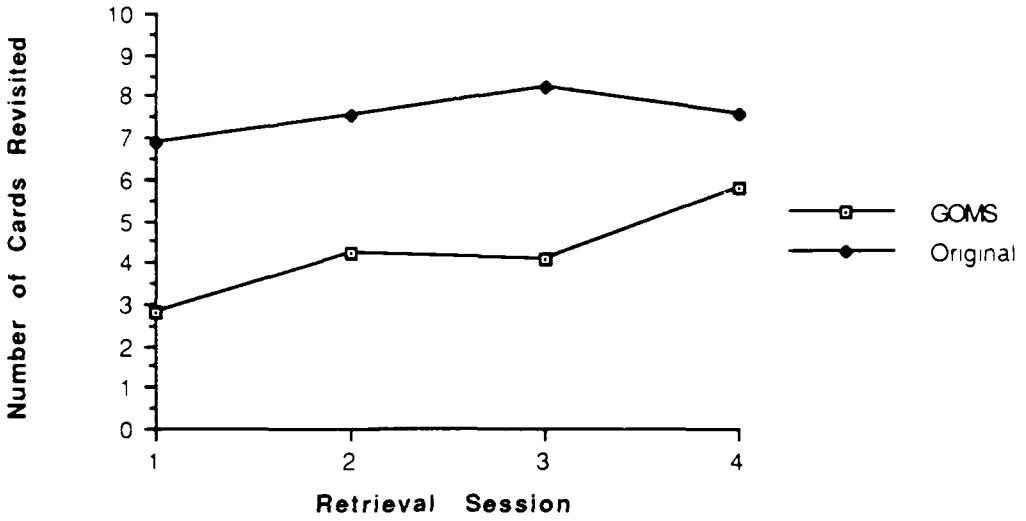


Figure 11. *Mean number of cards from previous trials revisited in the help stacks across retrieval sessions.*

These results can be understood when considering the structure of the Original help stack. Help information for different actions or objects were often placed on the same card. Examples of this can be seen on Cards 4 and A in Figure 3. Card 4 illustrates that several actions (i.e., move, size, copy, and delete) on fields were explained on a single card. Similarly, Card A in Figure 3 shows that selecting different objects (i.e., pictures, text, fields, or buttons) was explained on a single card. This is in strong contrast with the structure of the GOMS help stack where individual methods for each goal, consisting of an action and an object, were presented on separate cards. Thus, GOMS users may access some of the same navigational cards during a trial, but eventually would access different informational cards for the different authoring tasks. This different structure could have led to the difference in the number of cards revisited between the two help stacks.

Help by object analysis. Another analysis which sheds light on the differences between help stacks was an analysis of retrieval time for authoring tasks on different HyperCard objects. The reason there may be differences in retrieval performance for objects is that the GOMS stack was constructed using an explicit analysis of the objects and actions (i.e., the goal structure) for the HyperCard authoring task.

A two-way ANOVA on retrieval time revealed significant main effects for the type of help stack (F [1,26] = 10.50, p < 0.005) and type of object (F [6,156] = 21.8, p < 0.0001). In addition, a significant Help Stack by Object interaction was found (F [6,156] = 4.07, p < 0.0001). As can be seen in Table 5, retrieval times were consistently longer for the Original help stack. However, there were some authoring tasks which required significantly more time to find the help information. These authoring tasks involved stacks, cards, and backgrounds and required from 49% to 83% more time to find help information in the Original help stack. This result is surprising since stacks, backgrounds, and cards are the basic building blocks of HyperCard.

TABLE 5

Retrieval Time for the Different Objects in the Authoring Tasks in Each Help Stack

Object	GOMS	Original	Percentage Difference	Significant Difference?*
Stack	88.4 s	131.6 s	49 %	yes
Card	46.1	84.2	83	yes
Background	86.4	130.7	51	yes
Button	62.2	72.3	16	no
Field	70.8	76.8	8	no
Text in Field	61.1	73.0	19	no
Text in Picture	73.3	89.7	22	no

*Based on *post hoc* Newman-Keuls procedures (α = 0.05).

Help by action analysis. A similar type of analysis was conducted for authoring tasks which included different actions. This analysis would indicate the difficulty of finding

help information for different actions in the authoring tasks for the two help stacks. A two-way ANOVA on retrieval time revealed significant main effects for the type of help stack (F [1,26] = 10.24, p < 0.005) and type of object (F [6,156] = 23.87, p < 0.0001). In addition, a significant Help Stack by Object interaction was found (F [6,156] = 13.33, p < 0.0001). The data for this interaction are shown in Table 6.

As shown in Table 6, there was only one action in which the GOMS help stack yielded slower times when compared to the Original help stack. For the other actions it took users of the Original help stack from 3% to 154% longer to find information on specific actions. In fact, for the authoring tasks involving the actions of copy, move, and select there was a significant increase in the amount of time required to find the information in the Original help stack. This increase was also surprising since copy, move, and select are common actions used in HyperCard authoring tasks.

TABLE 6

Retrieval Time for the Different Actions in the Authoring Tasks in Each Help Stack

Action	GOMS	Original	Percentage Difference	Significant Difference?*
Check	90.4 s	86.5 s	- 4 %	no
Copy	56.7	89.3	57	yes
Create	88.2	105.6	20	no
Delete	44.4	52.5	18	no
Move	49.4	70.9	44	yes
Select	51.6	131.3	154	yes
Specify	92.0	94.9	3	no

*Based on *post hoc* Newman-Keuls procedures (α = 0.05).

Help by task type analyses. In addition to analyzing tasks with respect to the actions and objects that were involved, tasks were analyzed with respect to whether they share help cards and whether they had a similar procedural structure. For reasons explained earlier, we expected that tasks sharing cards or having similar procedural structure would lead to faster retrieval of help information with the GOMS help stack.

An initial analysis of the effect of task type on retrieval time for the two help stacks was performed using a two-way ANOVA. Significant main effects were found for the type of help stack (F [1,26] = 5.56, p < 0.05) and task type (F [3,78] = 65.34, p < 0.0001). In addition, a significant Help Stack by Task Type interaction (F [3,78] = 21.68, p < 0.0001) was found. As can be seen in Table 7, the largest significant difference between task types was for unique tasks. This large effect is consistent with the previous help by session analyses since many of the unique tasks were presented during the first session. This suggests that tasks not sharing cards or similar procedural structure in the GOMS help stack were much more difficult to find in the Original help stack. However, evidence for the superiority of the GOMS help stack for tasks sharing cards and with similar procedural structure was not found.

TABLE 7

Retrieval Time for the Different Task Types in Each Help Stack

Task Type	GOMS	Original	Percentage Difference	Significant Difference?*
Unique	73.6 s	125.5 s	71 %	yes
Shared Cards	75.4	75.9	0	no
Same Cards	59.5	60.0	0	no
Similar Method	46.5	55.0	18	no

*Based on *post hoc* Newman-Keuls procedures (α = 0.05).

Why was the type of task effect only partially observed? One possible explanation is related to the previous hypothesis on how the structure of the Original help stack encouraged users to revisit help cards. Since help information for different actions or objects were often placed on the same card, this reduced the users' search for new help information considerably. One would expect that this reduction in search for the Original help stack could offset any performance improvements due to the explicit structure of the GOMS help stack. Apparently, the Original help stack had a structure which made it easy to find related help information. However, the price paid was the initial time to find help information in the first session. Since

this initial analysis produced a generally uninteresting result for task type (i.e., we were not directly interested in the structure of the Original help stack), further analyses were not conducted.

Retrieval Accuracy

The accuracy of users retrieving the correct help information was analyzed at two levels. At a high level, these analyses determined the number of times users received different feedback messages (found all, some, or none of the help information) and how many cards were marked correctly or incorrectly. At a finer level, each card in both help stacks was scored for how many NGOMSL steps it contained. Then a detailed analysis was conducted on how much correct help information was retrieved.

Feedback scores. Table 8 shows the number of times users in both stacks were presented with the three types of feedback. In this analysis, feedback indicating that the user found all the information actually means that the user found all available information in the help stack. This is important since the Original help stack frequently did not have a complete method for each authoring task. As shown in Table 8, users of the GOMS and Original

TABLE 8

Feedback Frequency for the Users of the Two Help Stacks

	Type of Feedback			
Help Stack	None	Some	All	Total
GOMS	113	196	475	784
Original	157	157	470	784
Total	270	353	945	1568

help stacks received feedback that they found all the help information at approximately the same frequency. The major difference between the two groups was whether they retrieved some of the help information or none of the help information. When compared to Original users, GOMS

users had more instances in which they retrieved some of the information and fewer instances in which they retrieved none of the information. A chi-square analysis on the counts in the Table 8 revealed that the difference between GOMS and Original users was significantly different ($\chi^2[2] = 11.5$, $p < 0.01$). Although not a large practical difference, this analysis does reveal that users of the GOMS help stack found more partial help information than none at all.

Number of marked cards. The number of correct cards marked by users of both stacks was analyzed with an ANOVA. This analysis only revealed a significant main effect for retrieval session (F [3,78] = 34.47, $p < 0.0001$) with no significant effect for type of help stack (F [1,26] = 1.67, $p > 0.2$) or the Help Stack by Retrieval Session interaction (F [3,78] = 1.06, $p > 0.3$). As shown in Figure 12, there was very little difference between the two groups in each of the sessions.

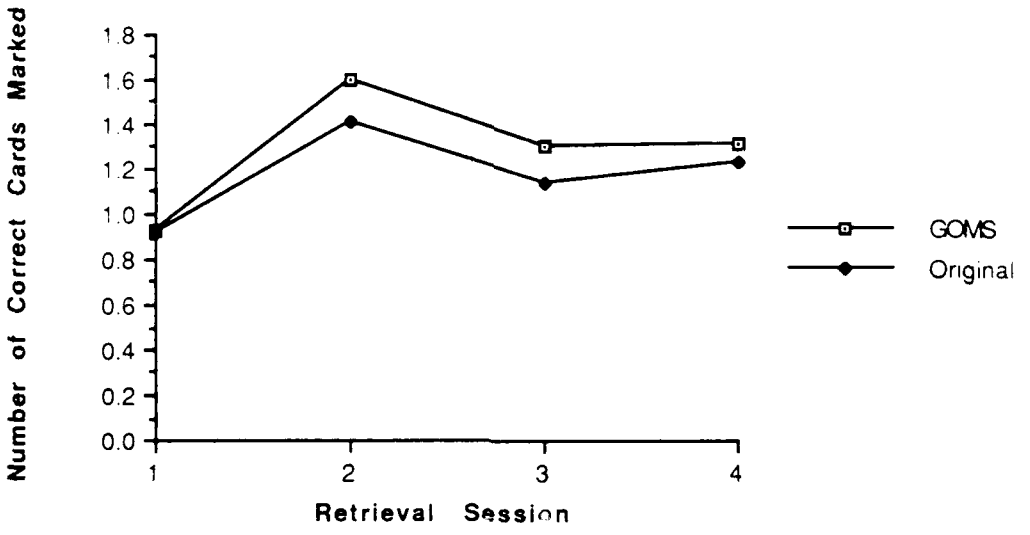


Figure 12. *Mean number of correct cards marked across retrieval sessions.*

A more interesting result was the number of incorrectly marked cards. An ANOVA revealed significant main effects for the type of help stack (F [1,26] = 4.24 , $p < 0.05$),

retrieval session (F [3,78] = 7.55, p < 0.0002), and the interaction between the type of help stack and retrieval session (F [3,78] = 11.23, p < 0.0001). As illustrated in Figure 13, *post hoc* Newman-Keuls procedures (α = 0.05) found that GOMS and Original users were marking approximately the same number of incorrect cards during the first retrieval session. However, after the first session there was a slight decrease in the number of incorrect cards marked for both groups followed by a significant increase in the GOMS group and an insignificant increase in the Original group. For sessions 2 through 4, GOMS users marked significantly more incorrect cards than Original users suggesting that GOMS users might be using different criteria for marking cards.

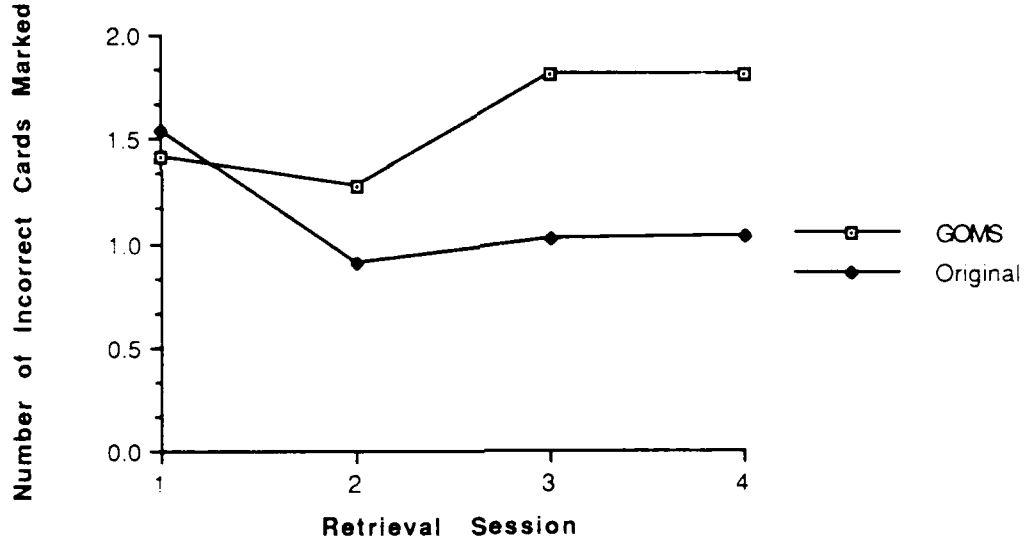


Figure 13. *Mean number of incorrectly marked cards across retrieval sessions.*

A detailed analysis of the location of the marked incorrect cards was conducted to determine if users were marking cards directly adjacent to and accessible from correct cards because they were unsure of whether to include these cards. This might increase the number of incorrect cards marked in the GOMS stack since many related help cards were linked together in

a hierarchical fashion. For example, in Figure 2 on Card 3 there is an asterisk indicating that more information was available on selecting fields (Card 4). Depending on how the authoring task was worded users may need to mark Card 4 in addition to Card 3. Therefore, if users initially did not mark all the required cards, they would receive feedback indicating that they had marked only some of the cards. This feedback could bias users to mark more cards than necessary for the authoring task. In contrast, this phenomenon would be less likely to occur with the Original help stack since cards are usually not linked in a hierarchical fashion.

An ANOVA on the number of incorrect cards which were marked incorrectly, but which were directly adjacent to a correct card, showed significant effects for the type of help stack (F [1,26] = 30.49 , p < 0.0001) and the Help Stack by Retrieval Session interaction (F [3,78] = 46.27 , p < 0.0001). No significant main effect for retrieval session was found (F [3,78] < 1.0 , p > 0.5). As illustrated by Figure 14, the data supported the hypothesis that after the first session GOMS users marked incorrect cards which were adjacent to correct cards more frequently than Original users. *Post hoc* Newman-Keuls procedures (α = 0.05) revealed that

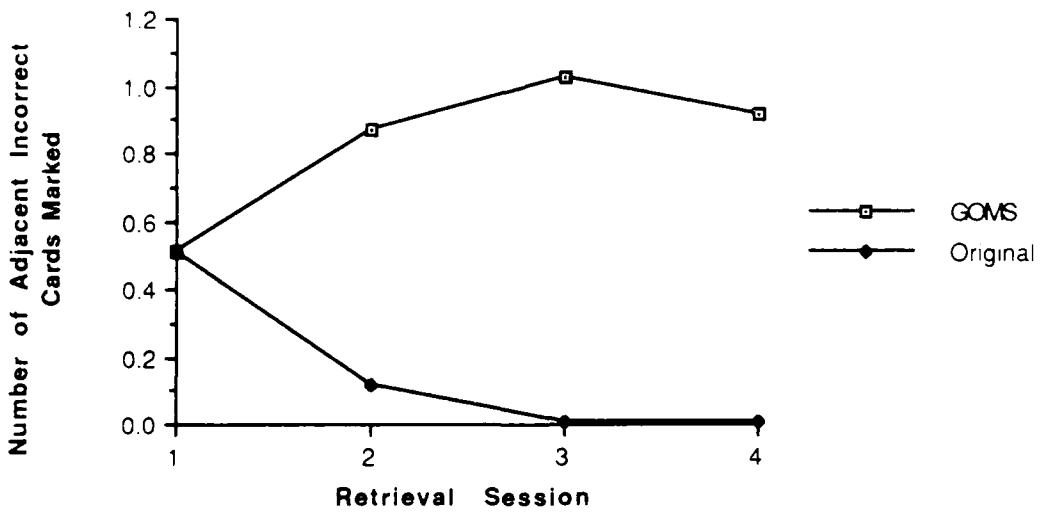


Figure 14. *Mean number of incorrect cards marked that were directly adjacent to a correct card for the help stacks across retrieval sessions.*

GOMS users significantly increased their rate of marking these cards after the first session while Original users significantly decreased marking these cards. In addition, these procedures showed that there were significant differences in the number of incorrect cards marked between GOMS and Original users for every session except the first. In fact, the difference of approximately 0.8 to 1.0 card in Sessions 2 through 4 between the GOMS and Original users (see Figure 12) is very close to the number of additional incorrect cards marked by GOMS users overall. This strongly supports the hypothesis that GOMS users were encouraged by the feedback and the linkage of the cards to mark cards directly adjacent to correct cards.

Number of NGOMSL steps. A finer level of accuracy analysis was conducted to determine how many observable NGOMSL steps were retrieved by users for each task. This required that the help cards for each task be scored in each stack to determine the number of observable NGOMSL steps. Scoring the GOMS stack was straightforward since the step-by-step directions were developed directly from the NGOMSL model. All that was required was to determine whether a step involved an observable operator. Examples of observable operators include selecting a command, clicking on a button, or pressing a key.

Scoring the Original help stack for the number of observable steps was more complicated. To score the Original help stack the two authors made independent judgements to determine the number of observable NGOMSL steps described on each card. A check on the reliability yielded a correlation coefficient of 0.89 and an agreement for the exact score of 71.5%. Appendix C details the scoring procedures used in this analysis. Based on this analysis, it was determined that the Original help stack had on the average 85.9% of the observable NGOMSL steps that were contained in the GOMS help stack. This confirms the assertion that the Original help stack was less complete than the GOMS help stack.

An ANOVA on the proportion of observable NGOMSL steps retrieved revealed significant main effects for the type of help stack (F [1,26] = 8.07, p < 0.01) and retrieval session (F [3,78] = 3.78, p < 0.05) with an interaction between the type of help stack and retrieval

session (F [3,78] = 5.09, p < 0.005). As can be seen in Figure 15, there was an overall difference in accuracy between the two groups (GOMS: 0.68 vs. Original: 0.59). However, the interaction shown in Figure 15 also shows that this effect was mediated by the retrieval session. *Post hoc* Newman-Keuls procedures (α = 0.05) revealed that users of the GOMS help stack retrieved significantly more NGOMSL information in Sessions 1 and 4, while differences between groups in Sessions 2 and 3 were insignificant. Depending on the retrieval session, users of the GOMS help stack retrieved approximately 5 to 20% more of the observable NGOMSL steps than users of the Original help stack. This was not the large superiority in retrieval accuracy that was predicted for GOMS help. Similar to the analysis at the feedback level, the difference in accuracy is statistically reliable, but it is questionable whether it is a practical difference. In fact, this observed difference in retrieval was approximately the same as the 14.1% average difference in the information available between the help stacks. Therefore, the GOMS help stack did not improve retrieval accuracy beyond what would be expected *a priori*.

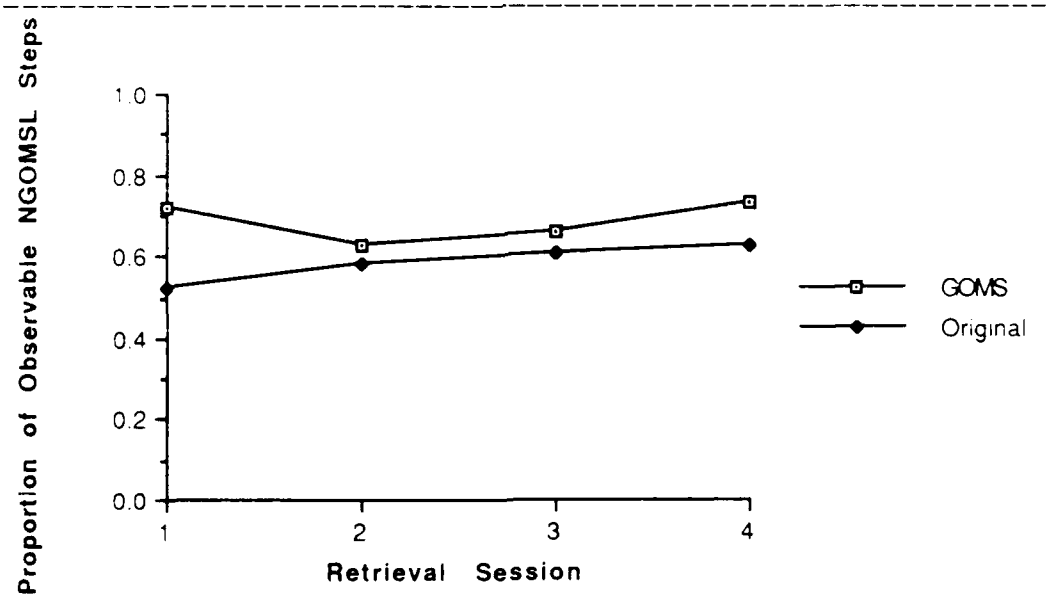


Figure 15. *Mean proportion of observable NGOMSL steps retrieved in the help stacks across retrieval sessions.*

These results are surprising when the absolute values of the proportions are taken into consideration. Poor retrieval accuracy can be expected for the Original help stack since help material was often spread over several unconnected cards. However, this was not the case for the GOMS help stack. Yet, users of this help stack failed to retrieve approximately 25% of the GOMS information. Why didn't GOMS users retrieve more information when it was completely documented in the help stack?

One hypothesis to explain this poor retrieval accuracy with the GOMS help stack was that users failed to mark relevant help cards directly adjacent to a correctly marked help card. Since the GOMS help stack was hierarchically structured, users would sometimes have to make judgements whether additional help information was needed for an authoring task. For the 333 tasks in which GOMS users failed to mark all the correct cards, 144 times (43%) users failed to mark a relevant help card which was directly adjacent to a correctly marked help card. Thus, this failure to mark directly adjacent and relevant help cards accounts for some of the poor retrieval of GOMS users and suggests that users had some difficulty making decisions on what help information was relevant to the authoring tasks.

In contrast, this failure to mark directly adjacent help cards was observed infrequently for Original users (49 of the 544 tasks in which users did not retrieve all the help information, 9%). This lower frequency was probably the result of the structure of the Original help system since there were many more instances when a related help card was not directly connected to a marked correct card (321 of the 544 tasks in which users did not retrieve all the help information, 59%). As expected, this data suggests that a problem with the Original help stack was that related help cards were not connected which may have resulted in users not retrieving this information.

Subjective Evaluations

The GOMS help stack was rated superior to the Original help stack on 15 of the 16 subjective scales with the final scale resulting in a tie. Therefore, there was a clear subjective

preference for the GOMS help stack. Users' subjective feelings about the two help stacks were analyzed using Wilcoxon rank-sum tests. A summary of the significant differences are provided in Table 9.

TABLE 9

Significant Results of the Subjective Evaluations Comparing the Two Help Stacks

Question	Subjective Ratings GOMS	Original	W	Normal Approx. (Z-score)	P-value
Specific Questions					
1. In general, how many cards did you need to go through to find the information needed? (1-very few, 5-a lot)	3.07	3.71	166.50	-1.74	0.0823
2. How often did you feel lost when using the HyperCard Help system? (1-never, 5-often)	2.36	3.07	162.50	-1.94	0.0528
3. Would you have preferred to have looked for the same information as was contained in the online HyperCard Help stack in a manual? (1-not at all, 5-very much)	1.64	2.71	155.00	-2.28	0.0226
General Impressions of HyperCard					
4. 1-simple, 5-complex	2.50	3.29	167.00	-1.70	0.0897

As shown in Table 9, Original users felt they browsed more cards in the help stack than GOMS users. Although this is not surprising since Original users actually did browse through more cards (see Figure 4), it does indicate that users were aware of how much information they looked through. More interestingly, Original users felt they were lost more often than GOMS users. The number of users selecting each point on this subjective scale is shown in Figure 16. As illustrated, more GOMS users were tending to indicate that they never felt lost, while the responses for the Original users were more evenly distributed across the scale. Apparently the

explicit structure of the GOMS help stack was useful in helping users navigate through the help information and preventing them from feeling lost.

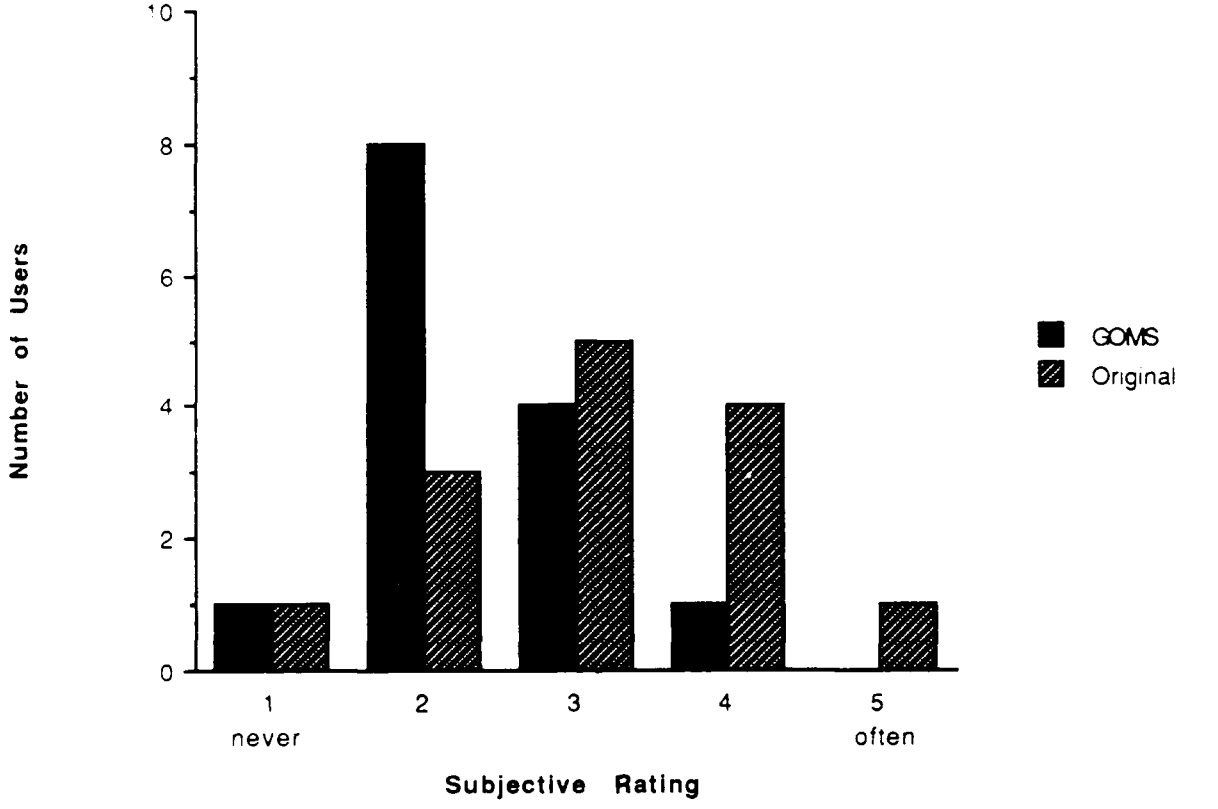


Figure 16. *Responses to the question: "How often did you feel lost when using the HyperCard help system?"*

Also shown in Table 9 is the significant subjective difference between GOMS users and Original users in terms of their preference for whether this help information should be placed in a manual or online. Figure 17 shows that the majority of the GOMS users indicated that they did not wish the help information to be contained in a manual. In contrast, Original users were more ambivalent on the implementation of the help. This may be interpreted that GOMS users were satisfied with the online help stack, whereas Original users were less satisfied and thought that there could be alternative ways of presenting the help.

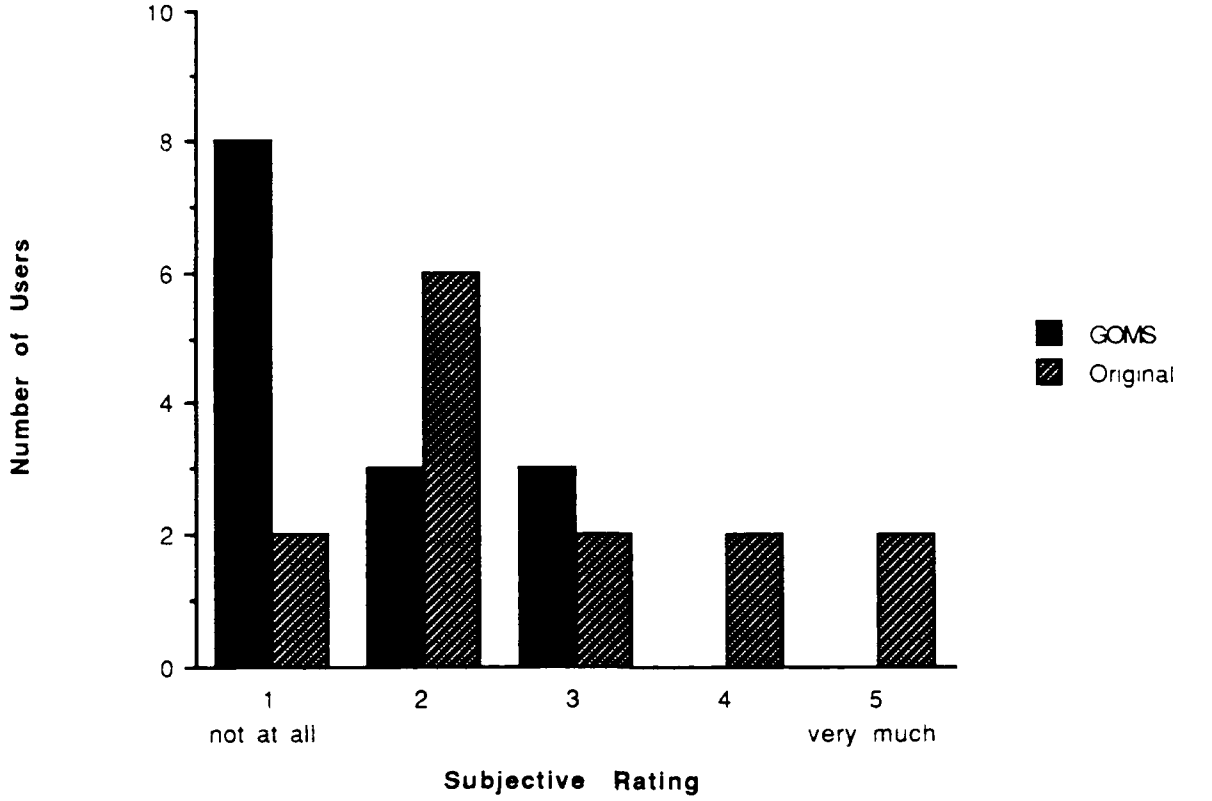


Figure 17. *Responses to the question: "Would you have preferred to have looked for the same information as was contained in the online HyperCard help stack in a manual?"*

Finally, the data in Table 9 indicate that users of the GOMS help stack thought that HyperCard was simpler than users of the Original help stack. As illustrated in Figure 18, 6 GOMS users were tending to indicate that HyperCard was simple, while only 3 Original users indicated this and none of these users indicated this strongly. Instead, 6 Original users tended to indicate that HyperCard was complex. Perhaps the GOMS help stack conveys the design of HyperCard in a straightforward way such that users feel that it is a less complex system.

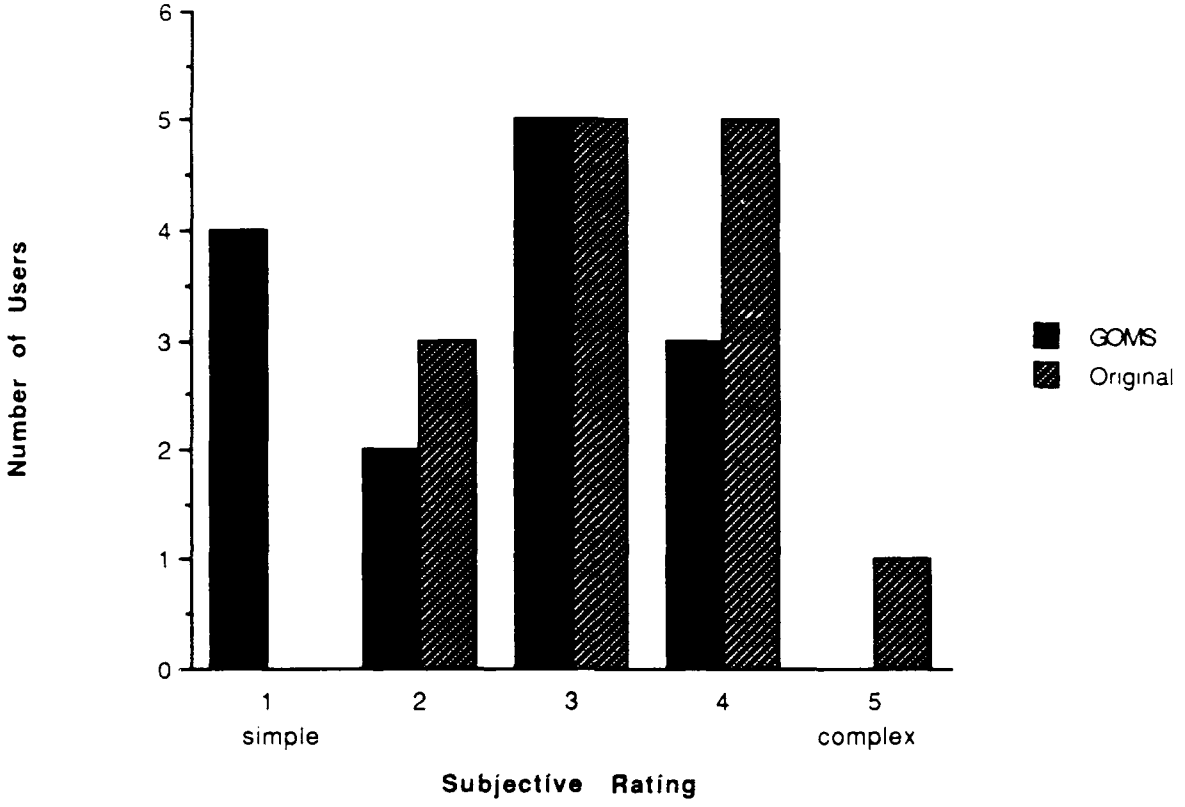


Figure 18. *General impressions of HyperCard for users of the two help stacks.*

DISCUSSION

Usability of the Help Stacks

Retrieval time and effort. The results indicate that the GOMS help stack was superior in terms of time and effort. However, this superiority was limited to the first session of the experiment. At least two possible hypotheses can explain these results. First, the additional time and effort may have been needed by the Original users to learn where help information was located in the help stack, while this did not seem to be necessary for the GOMS users. Related to these results is the hypothesis that the structure and content of the GOMS help stack was organized explicitly around HyperCard authoring tasks enabling users to determine where information was located.

The other explanation of the results is that the difference observed between GOMS and Original users may be due to the specific order of the authoring tasks. This explanation must be considered since the tasks were not counterbalanced between retrieval sessions, but were ordered pedagogically so that the simpler authoring tasks appeared before more complex tasks. This explanation does not rule out the previous, more interesting hypothesis. However, it does affect the generalizability of the results. If task ordering does play a factor in the usability of the two help stacks, then superiority of the GOMS stack can only be assumed for the pedagogical ordering used in this study. What is asserted here is that this experiment provides convincing support for the first explanation and also argues against the task-order interpretation.

One of the remarkable differences between the two groups was the strong learning effects observed with the Original help stack and the apparent lack of learning with the GOMS help stack. GOMS users only needed a fixed amount of time (approximately 70 s) to find help information for each authoring task. This constant retrieval time can be understood when considering the structure of the GOMS help stack. Help information in this stack was very modular. Individual help methods were provided on separate cards which were explicitly addressed by the goals of the authoring task (e.g., actions and objects). Users would first have to identify the appropriate action, then the appropriate object, and then they would be positioned at the correct help method for that task. This idealized retrieval behavior is quite similar for each authoring task, and consequently, explains the constant amount of time required for each retrieval of help.

Retrieval of help from the Original help stack, however, was probably much different. Initially, users had to explore many more help cards which may have allowed them to understand the structure and content of the help stack. The most dramatic data related to this initial exploration is the time required to find and mark the first piece of correct help information. Original users required almost triple the time to find correct help information in the first retrieval session. As noted, this could be quite frustrating for Original users. In the real world a help system requiring almost 2 minutes to find the first piece of help information

may not be used. Put another way: Why should users have to learn the structure of a help system in addition to learning about the specific help topic?

A closer look at the retrieval behavior of users explains even more. In addition to the extra exploration during the first session, Original users had to access more navigational cards during the entire experiment and never approached the level of GOMS users. Instead, comparison of Figures 8 and 9, illustrates that Original users increased their rate at which they accessed the navigational cards during Sessions 2 - 4. This may indicate that users were familiar with the stack structure and increased their rate to get through these additional navigational cards. Or, it could also mean that users were still unfamiliar and possibly confused with the stack structure and needed to access more navigational cards to find the appropriate help information.

Evidence for the explanation that users became familiar with the structure of the Original help stack was provided by the number of cards which were revisited. Throughout the experiment Original users consistently revisited more cards than GOMS users. As explained earlier, this revisiting behavior may have been encouraged by the structure of the two help stacks. The Original help stack often explained several help methods on a single card (see Card 4 and A in Figure 3), whereas the GOMS help stack was modular and encouraged users to access different help cards. Although this revisiting may be an indication of user unfamiliarity and possible confusion with the stack structure, the tremendous learning which Original users exhibited argues against this hypothesis. Thus, Original users were consistently and intentionally going back to the same help cards.

Further evidence for the difficulties users of the Original help stack had with its structure was revealed by the action and object analysis of the authoring tasks. These analyses indicated that Original users had more difficulty finding certain actions and objects. Specifically, Original users took longer to find help information on authoring tasks concerned with the actions of selecting, copying, and moving, and with the objects of stacks, cards, and backgrounds. This supports the hypothesis that users had to learn the structure of the Original

stack since additional time was required for retrieving help information on these tasks. Furthermore, the results are contrary to the task-order effect since tasks dealing with these actions and objects were distributed over all four sessions. More interesting, is the fact that users had difficulty with these common actions and objects. Perhaps these common actions and objects were overlooked in the design of the Original help stack since a formal procedural analysis (i.e., GOMS) was not performed.

Finally, the results from the task type analysis indicated that Original users had difficulty finding help information on unique authoring tasks, while GOMS users had much less difficulty. This may suggest that the structure of the Original help stack made it difficult to find these initial authoring tasks, whereas the explicit structure of the GOMS help stack expedited retrieval performance. However, both Original and GOMS users were equally efficient in retrieving the same and similar help information. Thus, once the location of the help information was learned, Original users seemed to remember and retrieve the location of related help information just as quickly as GOMS users. This was probably due to the structure and the high level of revisiting associated with the Original help stack. The Original help stack placed related help information together which encouraged users to revisit this location and probably improved the user's memory for this information.

Retrieval accuracy. Overall, differences in retrieval accuracy between the two help stacks were less than expected. At both a high-level analysis of the feedback provided and a detailed analysis of the number of observable NGOMSL steps retrieved there were only modest differences between the Original and GOMS help stack (approximately 5 - 20% favoring the GOMS group). Although 20% sounds like a large difference, approximately 14% of this difference could be accounted for by *a prior* differences in accuracy between the two stacks. Moreover, this difference may have been induced from the experimental conditions. Specifically, users may not have been given sufficient feedback to improve their retrieval performance. Just telling users that they had found none, some, or all of the information and

not providing users a chance to see the correct help information limits their corrective behavior. Ideally, the most natural way of correcting user's retrieval accuracy would involve having users perform the help methods for the authoring tasks.

The accuracy analysis, however, did disclose that GOMS users may have difficulties making decisions on what information was necessary to perform the authoring tasks. Indications of this difficulty comes from data which shows users either marking or not marking cards directly adjacent to correctly marked cards in the GOMS help stack. First, GOMS users marked more cards incorrectly which were directly adjacent to a correctly marked card than Original users. Second, GOMS users often failed to mark correct cards which were directly adjacent to a correct card.

How could these two results occur simultaneously? The answer may be that GOMS users had difficulty identifying the correct level of detail from the hierarchical, procedural directions. For example, the large number of incorrectly marked cards could have been navigational cards directly above the detailed, step-by-step methods in the GOMS hierarchy. Of the 603 tasks in which users marked incorrect cards 39% of the time the cards were navigational. In contrast, the large number of correct help cards which were not marked in the GOMS stack could have been additional detailed methods that the main help method accessed (e.g., opening a dialogue box). In this case, of the 144 tasks in which GOMS users failed to mark correct cards, 57% of the time the unmarked cards were detailed informational cards situated below a correctly marked card in the hierarchy. This data suggests that users were having some difficulty deciding whether cards above or below a correct card in the hierarchy should be marked. This may be an inherent difficulty with the hierarchical and adjustable level of detail in the GOMS help stack or it could be an artifact of the experimental procedures. Once again, just having users retrieve help information may not provide enough feedback on what help information was needed.

A final interesting result revealed by the accuracy analysis is the low overall retrieval accuracy of both help groups in terms of the number of observable NGOMSL steps retrieved.

GOMS users failed to retrieve 32% of the help information, while Original users failed to retrieve 41% of the help information. Based on an analysis of the cards users failed to mark, it was found that GOMS users may have been inaccurate for some of the reasons described above. However, for Original users the retrieval accuracy was due to the lack of links between related help cards. The Original help stack often failed to cross-reference help cards which were procedurally related.

For example, for the authoring task where the line height of text on a card must be determined, users would have needed to access both Card 1 and Card 2 in Figure 19 for perfect accuracy. Essentially, Card 1 tells the user how to open the dialogue box shown in Card 2 which displays the line height. However, there is no direct link between Card 1 and 2. This inadequacy of the Original help system may have resulted from a lack of formal analysis of the procedural structure of HyperCard authoring tasks. Such an analysis would have identified the procedures which should have been cross-referenced so that users could have easily accessed the additional related help methods.

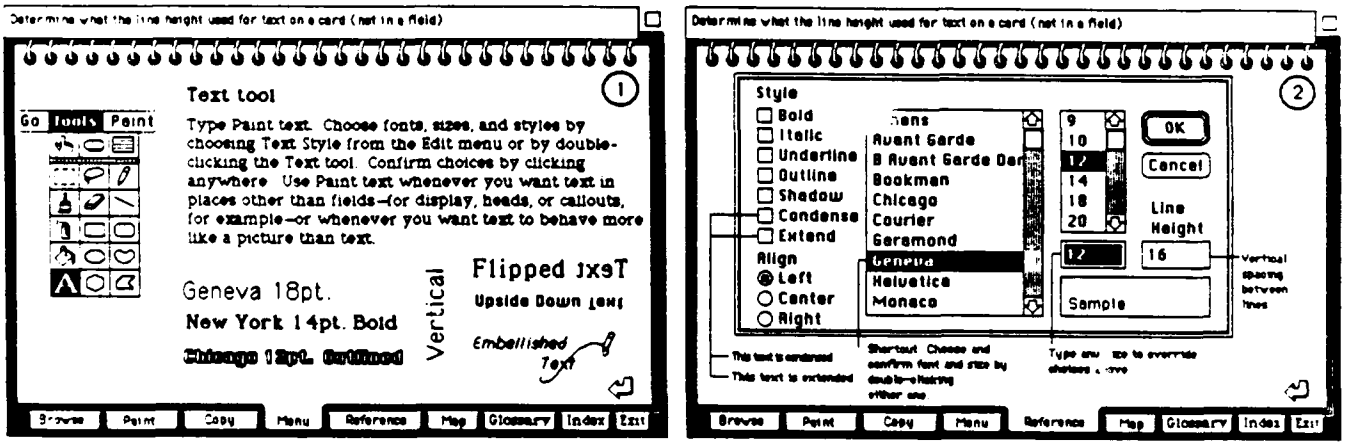


Figure 19. *Example of procedurally-related help cards from the Original help stack which were not linked together.*

Formal and informal subjective evaluations. In total, the GOMS help stack was preferred subjectively over the Original help stack. Users of the GOMS help stack felt they had to go through less help information, felt lost to a lesser extent, preferred the online format, and viewed the HyperCard application as more simple than complex. These formal evaluations also agreed with the informal comments users provided to an open-ended questionnaire. GOMS users often commented that the help information was structured effectively for fast and efficient performance. Although there were few negative comments about the usability of the GOMS help stack, some users stated that sometimes they felt lost and that they disliked going through several layers of help information.

Similarly, the informal comments for the Original help stack paralleled the formal evaluations. Some users commented that the Original help stack should be restructured. One user commented that the help stack should be more task-oriented and another user made the specific suggestion that the help stack should be organized around the interface objects. A few Original users stated that the index was helpful with one user adding that the index should be redesigned to increase its use. Lastly, some users noted that it took time to understand where information was located in the Original help stack.

Clearly, these subjective evaluations closely match the observed retrieval performance. Users apparently liked the structure of the GOMS help stack, whereas users of the Original help stack seemed to be frustrated with its structure. Interestingly, users of the Original help stack suggested changes for the Original help stack which seemed largely responsible for the success of the GOMS help stack. Apparently, some of the improvements in the GOMS help stack are exactly what users want.

Summary. This information retrieval study showed that using a GOMS model to design a help system can improve users' retrieval performance and satisfaction with the online help. However, the results of this experiment may only be a conservative estimate of the superiority of the GOMS help stack for two reasons. As mentioned earlier, five participants were excluded

from the experiment due to technical problems. As it turned out, these five participants were all assigned to the Original help stack. Three of the five were excluded due to software failures which were caused by data overflow since these users searched the help stack extensively and took a great deal of time to find the help information. The other two participants had to stop in the middle of the study due to personal time constraints since they also were taking longer than expected to find the help information. Therefore, these five participants were very slow retrieving help and if included in the Original help group would have increased the differences in time and effort.

A second reason for the results being a conservative estimate of the differences between the two help stacks is related to the experimental procedure employed in the study. In the experiment, users were required to repeatedly find help information. This probably reduced the time and effort differences since users were able to focus solely on the retrieval task. If users would have to find the help information and also perform the authoring task there might be larger performance differences between the GOMS and Original help stacks.

Pros and Cons of Using GOMS for Help System Design

Benefits. The use of GOMS models to develop help systems has benefits which result in improved user performance and satisfaction with the system. To reiterate a previous point, using GOMS models allows the designer to structure the help system in relation to the user's tasks. This focus on the tasks decreases the amount of learning associated with help system, thereby allowing users to focus on learning the computer task for which they seek help. This seemed to have taken place for the GOMS help stack. These users took a constant amount of time to find help information indicating that there was little learning associated with the structure and location of help. In contrast, users of the Original help stack needed initial learning time to understand the structure and location of help information.

A large part of the effective structure of the GOMS help stack was developed by using the goal hierarchy to serve as an index into the detailed step-by-step methods. This provides

explicit cues to users about what actions on what objects (i.e., goals) were possible. Unlike the Original help stack, users did not have to guess where information was located. In addition, this goal structure allowed relatively quick access to the detailed methods.

Using a goal structure to organize user documentation is not new. Carroll and his colleagues (Carroll, Mack, Lewis, Grischkowsky, and Robertson, 1985; Carroll, Smith-Kerker, Ford, and Mazur-Rimetz, 1988) have used user goals and tasks to organize material in manuals and user-oriented reference cards. What is new is the use of a formal model such as GOMS to develop this organization Instead, Carroll, et al. (1985; 1988) used empirical user testing to suggest the goals and tasks. Although this is one method of generating user goals (and perhaps a good first start), it is potentially time consuming and does not guarantee that all the possible goals and tasks will be observed. A formal GOMS model provides a method for analyzing the user's task more completely. This is important since only high-level guidelines (Carroll, et al., 1988) and advice (Brockmann, 1986) have been provided to make user documentation more task-oriented.

In addition to focusing on tasks, use of the GOMS model allows other principles of developing good user documentation to be followed. For example, Carroll, et. al. (1988) state that one important principle for designing a minimal manual is to slash the verbiage. How does one do this? Use of a GOMS model to focus the documentation or help on the procedures necessary to accomplish tasks is a promising approach. In this study, the size of the GOMS help stack was 25% smaller than the Original help stack (GOMS: 175 cards vs. Original 233 cards).

Beyond providing a method to adhere to these principles, the use of GOMS allows the designer to assess the completeness and consistency of the help system. Completeness of the help instructions can be assessed since GOMS is a model that can be executed by a human. That is, the analyst can test the completeness by hand simulating task procedures. This quickly identifies missing steps in the help instructions. Furthermore, if NGOMSL (Kieras, 1988) is used, a computer simulation of the procedures in the form of a production system can be developed (see Anderson, 1983; Bovair, Kieras, and Polson, 1988). Although computer

simulations were not run in the present experiment, this approach would provide an even more rigorous test of the explicitness and completeness of the help instructions.

In the present study, there were differences in the completeness of the directions in the two help stacks. The Original help stack had approximately 85% of the total observable NGOMSL steps contained in the GOMS help stack. Although this did not seem to lead to any further decrements in the amount of help information retrieved, the completeness of the help directions may have a definite impact on whether people can execute (perform) the help methods. Indeed, the philosophy when using GOMS for designing online help is that the described procedure should be complete. This is at odds with other efforts (i.e., Carroll, et al., 1985; 1988) where incomplete directions are thought to be beneficial because of their capability to engage the user to learn the task actively. The contention of the GOMS approach is that reference materials must be complete since you cannot predict the inferences that will be made with incomplete help information.

One final benefit of using GOMS for designing online help is one of consistency. At the very lowest level, consistency in the wording of help instructions can be enforced by translating the GOMS model to written instructions in a routine fashion. At a higher level, GOMS and production rule analyses are ideal for identifying general methods that can be used in different contexts (see Catrambone, in press). Identifying and describing these general methods could aid the user in learning about interface consistency.

Problems. One of the most prevalent problems with hypertext systems is that of getting lost (Conklin, 1987). Even though more users of the Original help stack indicated that they felt lost, some users of the GOMS help stack indicated that they felt lost. Why do people feel lost in the GOMS help stack? One explanation may be the depth of the goal hierarchy. The depth of the goal hierarchy for the GOMS help stack ranged from 1 to 9 cards deep with an average depth of 3.5 cards. Some depth in the goal hierarchy may be necessary to convey the procedural structure of the task. However, there may be a tradeoff between this benefit and the possibility

of users becoming lost. One possible solution to users getting lost in deep goal hierarchies may be to write longer and less hierarchical procedures. Another solution is to provide a fisheye view of the goal hierarchy (Furnas, 1986) so that users could keep track of their high-level goals while browsing lower-level subgoals.

The depth of the goal hierarchy may also impede users from quickly accessing detailed help methods. Some limited evidence for this is that GOMS users on the average had to browse 9 cards during a trial and took 38.8 seconds to find and mark the first correct card. Although they accessed fewer cards and took less time than the Original users, this performance quite possibly could be improved. Once again, the solution may be one of making longer and less hierarchical procedures. For information retrieval, there are few benefits for deep hierarchies since additional goals have to be read and each goal accompanied by a button click to access the method for that goal. Thus, hierarchical help instructions will always lead to longer retrieval times.

The data on retrieval accuracy also indicates that there may be problems with hierarchical help instructions. Users of the GOMS help stack failed to mark 32% of the correct help information. As previously discussed, users seemed to have problems making decisions on what information was necessary to perform the authoring tasks. More specifically, users sometimes left lower-level help methods unmarked. Similar problems of users failing to consider lower-level methods were observed by Kieras, Tibbitts, and Bovair (1984). These researchers found that users of menu-based instructions for a device often failed to look at sub-menu instructions. This then resulted in longer performance times since users did not know how to execute tasks associated with these lower-level methods. In future studies, these problems in retrieving help information may lead to poor execution performance for users who are not well practiced on specific HyperCard authoring skills since they too may skip reading about lower-level skills.

Future Research

Since information retrieval is only part of the use of online help, other experiments are planned to evaluate the use of GOMS for help system design. In particular, a method execution study is planned to see how well users can perform the help methods described in the GOMS and Original help stacks. This study is to determine whether the complete, goal-structured directions of the GOMS help stack improve users' speed and accuracy when compared to the incomplete instructions in the Original help stack. Also of interest is whether GOMS users are inaccurate in performing authoring tasks due to their possible failure to consider lower-level methods.

A final usability evaluation involving retrieval of help information and execution of help methods is also planned. The goal of this study is to confirm that the differences observed in the information retrieval and execution experiments generalize to more realistic help scenarios. Instead of performing individual authoring tasks users will be asked to perform a high-level task, such as creating a stack, which will require users to retrieve and execute help methods for individual authoring tasks.

CONCLUSIONS

This experiment shows clearly that a GOMS model can assist in developing a help system which improves user information retrieval performance. The GOMS help stack was easy to use, easy to learn, and well liked for information retrieval. This is to be contrasted with the Original help stack which required additional time to learn the structure and location of help information. The key characteristic for this improved performance seems to be the goal-oriented, procedural structure of the GOMS model which is explicitly represented as an index for the step-by-step methods. Still, despite these benefits there are some challenges which have to be addressed in future research. Further research must determine how the intermediate goal structure of a task can be represented without adding information access time or getting users lost. In addition, ways to encourage users to find and access lower-level help methods

need to explored. Nevertheless, the positive results from this experiment suggests that GOMS should be added to the relatively sparse set of methods available to help designers specify help content.

TRADEMARKS

ACKNOWLEDGMENTS

This research was supported by the Office of Naval Research under contract number N00014-87-K-0740. The technical monitor was John J. O'Hare. The authors would like to thank David Kieras for his advice and Lisa DeLisle for running the study.

REFERENCES

Anderson, J. R. (1983). ***The architecture of cognition.*** Cambridge, MA: Harvard University Press.

Borenstein, N. S. (1985). ***The design and evaluation of on-line help systems.*** Unpublished doctoral dissertation, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Bovair, S., Kieras, D. E., and Polson, P. G. (1988). ***The acquisition and performance of text editing skill: A production system analysis*** (Technical Report No. 28). University of Michigan and University of Colorado.

Brockmann, J. R. (1986). ***Writing better computer user documentation.*** New York: Wiley.

Card, S. K., Moran, T. P., and Newell, A. (1983). ***The psychology of human-computer interaction.*** Hillsdale, NJ: Lawrence Erlbaum.

Carroll, J. M., Mack, R. L., Lewis, C. H., Grischkowsky, N. L. and Robertson, S. R. (1985). Exploring, exploring a word processor. ***Human-Computer Interaction, 1,*** 283-307.

Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., and Mazur-Rimetz, S. A. (1988). The minimal manual. ***Human-Computer Interaction, 3,*** 123-153.

Catrambone, R. (in press). Specific versus general procedures in instructions. ***Human-Computer Interaction.***

Cohill, A. M., and Williges, R. C. (1985). Retrieval of HELP information for novice users of interactive computer systems. ***Human Factors, 27,*** 335-344.

Conklin, J. (1987). Hypertext: An introduction and survey. ***Computer, 20*** (9), 17-41.

Duffy, T. M., Mehlenbacher, B., and Palmer, J. (in press). The evaluation of online help systems: A conceptual model. In E. Barrett (Ed.) ***The society of text: Hypertext, hypermedia, and the social construction of information.*** Cambridge, MA: MIT Press.

Dunsmore, H. E. (1980). Designing an interactive facility for non-programmers. In ***Proceedings of the ACM National Computer Conference*** (pp. 475-483). New York: ACM.

Elkerton, J. (1988). Online aiding for human-computer interfaces. In M. Helander (Ed.) ***Handbook of human-computer interaction*** (pp. 345-364). New York: North Holland.

Furnas, G. W. (1985). Generalized fisheye views. In ***Proceedings of CHI'86: Human Factors in Computing Systems*** (pp. 16-23). New York: ACM.

Houghton, R. C. (1984). Online help systems: A conspectus. ***Communications of the ACM, 27,*** 126-133.

Kearsley, G. (1988). ***Online help: Design and implementation.*** Menlo Park, CA: Addison-Wesley.

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.) ***Handbook of human-computer interaction*** (pp. 135-158). New York: North Holland.

Kieras, D. E., Tibbitts, M., and Bovair, S. (1984). ***How experts and non-experts operate electronic equipment from instructions*** (Technical Report No. 14). University of Arizona, Tucson, Arizona.

Magers, C. S. (1983). An experimental evaluation of on-line HELP for non-programmers. In ***Proceedings of CHI'83: Human Factors in Computing Systems*** (pp. 277-281). New York: ACM.

Relles, N. (1979). ***The design and implementation of user-oriented systems.*** Unpublished doctoral dissertation, University of Wisconsin, Madison, WI.

Shneiderman, B. (1986). ***Designing the user interface.*** Reading, MA: Addison-Wesley.

Wright, P. (1988). Issues of content and presentation in document design. In M. Helander (Ed.) ***Handbook of human-computer interaction*** (pp. 629-652). New York: North Holland.

APPENDIX A

Authoring Tasks Used in the Experiment

Shown below are the authoring tasks used in this experiment. Included with these tasks are the session that the task appeared in and the task classification. The classification consisted of tasks which required the use of: (1) completely unique help methods (UN), (2) methods that share help cards from previous tasks (ShC), (3) help methods with a procedural structure similar to help methods seen in previous authoring tasks (SM), and (4) help methods that are contained on exactly the same cards as previous authoring tasks (SaC). The number of the previous task(s) is presented when the task classification refers to other previous tasks.

Task Number	Task Type	Task
		Session 1
1	UN	Pick some text on a card (not in a field)
2	UN	Duplicate the current stack, but do not change the stack in any way
3	UN	Type text onto a card (not in a field) using the current text characteristics
4	UN	Pick a stack
5	UN	Relocate a card to another place in the stack
6	UN	Safeguard the current background so it cannot be deleted from the stack
7	UN	Make a card by picking a command from a menu, but do not change the card in any way
8	UN	Pick a background from the current stack
9	UN	Make an empty background in an existing stack
10	UN	Pick a button
11	UN	Make a stack with one empty background
12	UN	Duplicate a card and place it in another part of the stack
13	UN	Pick a field
14	UN	Determine what is the user level for the current stack
		Session 2
15	ShC-4	Duplicate a selected background and put it into another stack
16	ShC-24	Name a card
17	ShC-10,26, 19	Unlock a selected field
18	UN	Determine how many backgrounds are contained in the current stack
19	ShC-17	Relocate a selected button from one card to another card
20	ShC-14	Get rid of the current stack
21	ShC-28	Duplicate a selected field on a card and place it on another card
22	ShC-25	Connect a selected button to a card in the current stack

23	UN	Make a field on a card by picking a command from a menu, but do not change it in any way
24	ShC-16	Get rid of a card
25	ShC-22	Make an opaque button with a user-defined size for a card
26	ShC-17	Determine what is the font size of the text in a selected field
27	UN	Get rid of a selected button
28	ShC-21	Duplicate a selected field and place it on the same card by using the option key

Session 3

29	ShC-22	Pick text in an unlocked field
30	ShC-25	Change a selected button so that it will display the button name on the screen
31	SM	Duplicate selected text on a card (not in a field) and place it on another card
32	ShC-14,20	Set up the current stack so that a password must be used when it is entered again
33	ShC-26	Determine what is the line height used for text on a card (not in a field)
34	SM	Get rid of selected text from a card (not from a field)
35	UN	Type text into an unlocked field using the current text characteristics
36	ShC-6	Determine how many cards have the same background
37	SM	Get rid of selected text from a field
38	SM	Relocate a selected field from one card to another card
39	ShC-17,26	Center some existing text in a selected field
40	SM	Relocate selected text on a card (not in a field) to another card
41	SM	Get rid of a selected field
42	SM	Duplicate a selected button and place it on the same card by using the option key

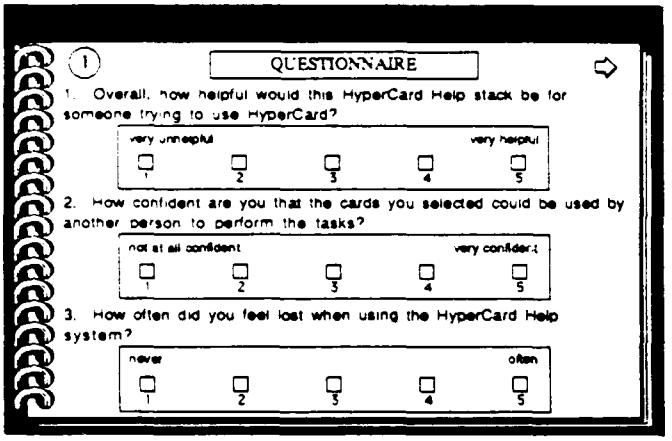
Session 4

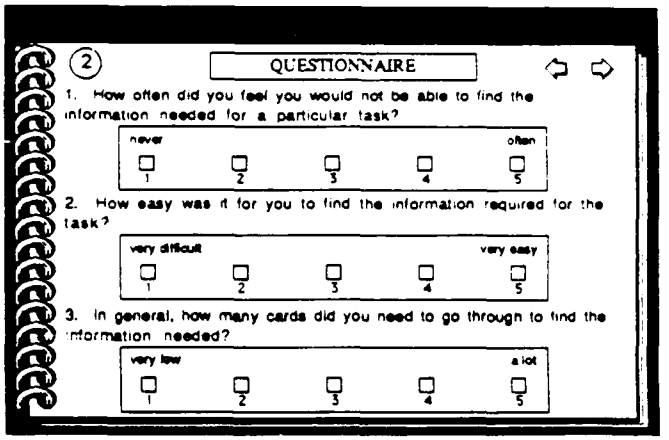
43	SaC-22,25, 30,47,51	Change a selected button so that it automatically highlights when used
44	UN	Relocate selected text in an unlocked field to another location in the same field
45	ShC-16,24	Determine what is the ID of the current card
46	ShC-26,33, 39	Alter the font for text to be added on a card (not in a field)
47	ShC-22,25, 30,43,47	Replace the current icon for a selected button with another icon
48	UN	Duplicate selected text from an unlocked field and place it in another location in the same field
49	SaC-19	Relocate a selected button to another position on a card
50	UN	Increase the dimensions of a selected field
51	ShC-22,30	Determine what is the number of a selected button
52	SaC-17,26, 39	Name a selected field
53	SaC-34	Get rid of selected text from a card (not from a field), but save a

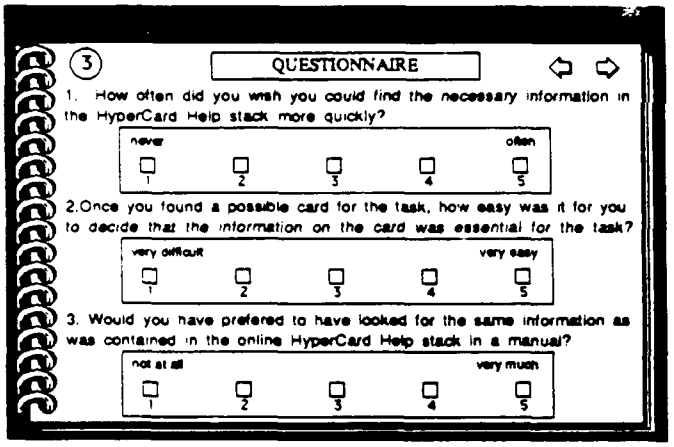
		temporary duplicate of it
54	SaC-17,38	Relocate a selected field so that it is on top of other objects on a card
55	SaC-27	Get rid of a selected button, but save a temporary duplicate of it
56	SaC-1	Pick text on a card (not in a field) that was just typed in

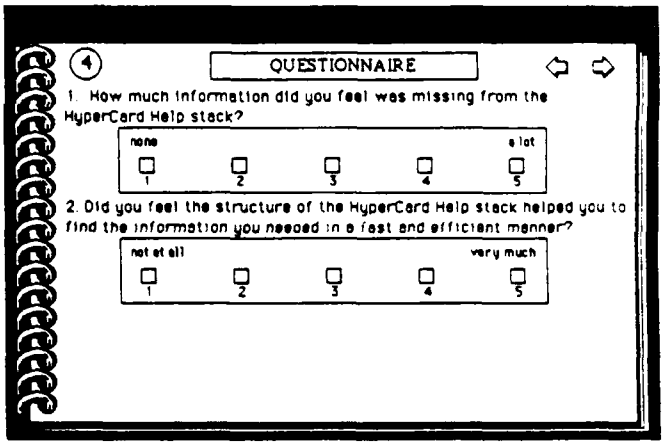
APPENDIX B

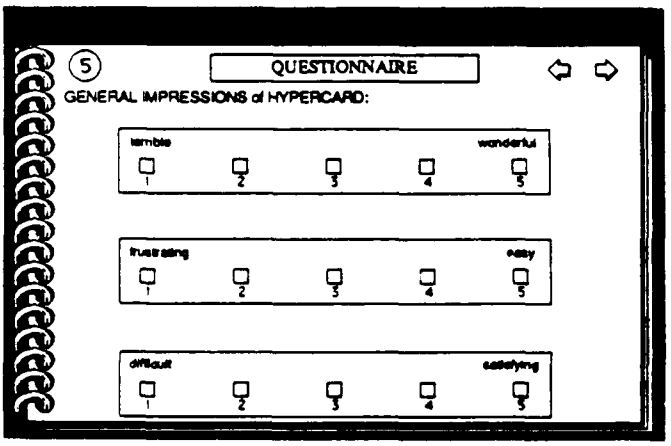
HyperCard Subjective Questionnaire

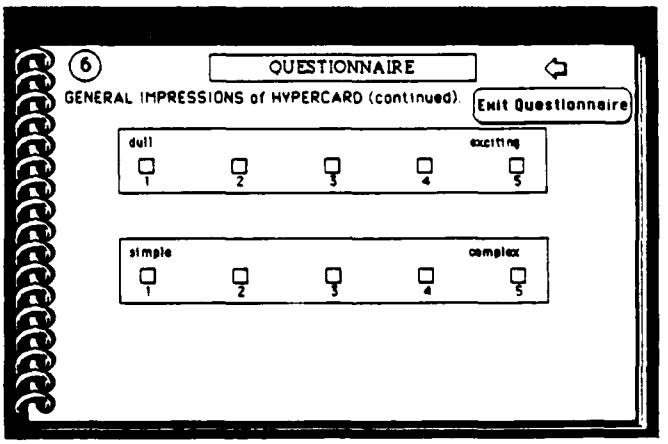












APPENDIX C

Scoring Procedures for Determining the Number of Observable NGOMSL Steps

To score the Original help stack the two authors made independent judgements on each card which had relevant help information in terms whether an observable NGOMSL step was described on the card. Cards with relevant help information were culled from the Original help stack by using string searches for words (or related words) in the authoring task. Scores consisted of 1, 0.5, and 0. A score of 1 indicated that the observable NGOMSL step was present on the help card. A lenient scoring scheme was used for this category such that a step was scored as present if it could be inferred by the user. A score of 0.5 indicated that half of a step was present. This scoring category was required since there were many help cards which had only a portion the observable NGOMSL step. An example of a half step is a help card which indicated the appropriate menu command, but failed to identify where it was located in the menu. Finally, a score of 0 indicated that the observable NGOMLS step was not present.

After initial scoring was complete a reliability analysis was conducted to determine how closely the two authors' scores matched. This initial scoring for the two authors resulted in a Pearson product-moment correlation of 0.76. A *t*-test of the coefficient for the slope of the regression indicated that this was a highly significant correlation (t [353] = 22.07, p < .0001). A comparison of the exact scores of each card revealed that there was 60.5% agreement between the two authors. Given the borderline reliability coefficient and the modest agreement between authors, a checking procedure was devised to see where there was divergence in the scoring of the two authors.

To check where the authors disagreed, regressions were run predicting one author's scores from another and vice versa. From each regression, standardized residuals were calculated to determine cards which had large relative disagreements between the two author's scores. Cards which had standardized residuals ±1.96 (studentized residuals which were in upper and lower 2.5% of the distribution) were chosen for further scoring analysis. Based on this residual analysis, differences were resolved between the two authors' scores. For the most part, differences between scores were the result of calculation errors, scoring inconsistencies for each author, and judgement differences for observable NGOMSL steps which were scored as half steps by one author and full steps by the other author.

The revised scores of each author were then reanalyzed in terms of reliability and agreement. The revised scores resulted in a Pearson product-moment correlation of 0.89. Once again, a *t*-test of the coefficient for the slope of the regression indicated that this was a highly significant correlation (t [353] = 37.62, p < 0.0001). A comparison of the exact scores of each card revealed that there was 71.5% agreement between the two authors.

Due to the higher reliability and agreement between the two authors' scores, one of the scoring schemes was selected to compare the marked cards of users to determine how many observable NGOMSL steps they had retrieved for each task. From the regressions of the two authors' scores it was found that one author was more lenient in scoring (first author's score = 0.0001 + 0.972 * second author's score). To give users the benefit of doubt, the more lenient scoring scheme was selected (the first author's scoring scheme).

Distribution List for This Report

OSD

Dr. Earl Alluisi
OUSDR(A)/R&A(E&LS)
Pentagon 3D129
Washington, DC 20301-3080

DEPARTMENT OF THE NAVY

Mr. Luis Cabral
Naval Underwater Sys. Ctr.
Code 2212
Bldg. 1171/1
Newport, RI 02841

Cdr. Robert C. Carter USN
Naval Research Laboratory
Code 5532
Washington, DC 20375-5000

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 North Quincy Street
Arlihngton, VA 22217-5000

Commanding Officer
Navy Personnel R&D Center
San Diego, CA 92152-6800

Commanding Officer
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, DC 20361

Commanding Officer
Navy Health Research Center
P.O. Box 85122
San Diego, CA 92138

Commanding Officer
Naval Biodynamics Lab
Michoud Station
Box 29407
New Orleans, LA 70819

Commanding Officer
Naval Weapons Center
Human Factors Branch, Code 3152
Naval Weapons Center
China Lake, CA 93555

Dean of the Academic Departments
U.S. Naval Academy
Crew Station Design
Annapolis, MD 21402-5018

Director
Technical Information Division
Code 2627
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Robert A. Fleming
Naval Ocean Systems Center
Human Factors Support Group
1411 South Fern Street
Arlington, VA 22202-2896

Jeffrey Grossman
Naval Ocean Systems Center
Code 4403, Bldg. 334
San Diego, CA 92152-6800

Capt. Thomas E. Jones, MSC, USN
Aviation Medicine &
Human Performance (Code 404)
Naval Medical R&D Com
National Capital Region
Bethesda, MD 21814-5044

Mr. Keith Kramer
Naval Research Laboratory
Code 5532
Washington, DC 20375-5000

Dr. Michael Letsky
Office of the Chief of Naval
Operations (OP-01B7)
Washington, DC 20350

Lt. Dennis McBride
Human Factors Branch
Pacific Missile Test Center
Point Mugu, CA 93042

Mr. Harold G. Miller
Technical Director, War
Gaming Department
Naval War College
Newport, RI 02841

Capt. W. Moroney, USN
Naval Air Development Center
Code 602
Warminster, PA 18974

Naval Aerospace Medical
Research Laboratory
Sensory Division, Code 23
Pensacola, FL 32508

Dr. A.F. Norcio
Computer Sciences & Systems
Code 5592
Naval Research Laboratory
Washington, DC 20375-5000

Office of the Chief of Naval Operations
OP-933D3
Washington, DC 20350-2000

Office of Naval Research
Perceptual Science Program (3 copies)
Code 1142 PS
800 North Quincy Street
Arlington, VA 22217-5000

Dr. W.A. Rizzo
Head, Human Factors Division
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3224

Dr. Randall P. Shumaker
Naval Research Laboratory
Code 5510
U.S. Navy Center for ARAI
Washington, DC 20375-5000

LCdr Timothy Singer, USN
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. James G. Smith
Office of Naval Research
Code 121
800 N. Quincy Street
Arlington, VA 22217-5000

Capt. Frank Snyder, USN
Naval War College
Operations Department
Newport, RI 02841

Cdr. S. Snyder
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Prof. Michael Sovereign
Naval Postgraduate School
Code 74
Joint C3 Curriculum
Monterey, CA 93943

Office of Naval Research
Special Assistant for Marine
Corps Matters
Code OOMC
800 North Quincy Street
Arlington, VA 22217-5000

U.S. Naval Test Center
Aircrew Systems Branch
Systems Engineering Test Directorate
Patuxent River, MD 20670

DEPARTMENT OF THE ARMY

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Michael Kaplan
Director, Office Basic Res
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. John Weisz
Technical Director
U.S. Army Human Engineering Laboratory
Aberdeen Proving Ground, MD 21005

DEPARTMENT OF THE AIR FORCE

Dr. Charles Bates, Director
Director, HE Division
USAF AAMRL/HE
Wright-Patterson AFB, OH 45433

Dr. Kenneth R. Boff
AAMRL/HE
Wright-Patterson AFB, OH 45433

Dr. Alfred R. Fregly
AF Office of Sci. Res.
Life Sciences Directorate
Bldg. 410
Bolling AFB, DC 20332-6448

Mr. Yale Smith
Rome Air Development Ctr.
RADC/COAD
Griffis AFB, NY 13441=5700

OTHER GOVERNMENT AGENCIES

Dr. Bruce Barnes
Program Director, Div. Info
Robotics, & Intell. Systems
National Science Foundation
1800 G. Street, N.W.
Washington, DC 20550

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314
(2 copies)

Dr. Richard Loutitt
National Science Foundation
Division of Behavioral & Neural Sciences
1800 G. Street NW
Washington, DC 20550

Dr. Harold P. Van Cott
Committee on Human Factors
NAS-National Research Council
2101 Constitution Avenue, NW
Washington, DC 20418

OTHER ORGANIZATIONS

Prof. Richard Catrambone
Georgia Institute of Technolgy
School of Psychology
Atlanta, GA 30332

Dr. Marvin S. Cohen
Dec. Sci. Consortium, Inc.
1895 Preston White Drive
Suite 300
Reston, VA 22091

Dr. Thomas Duffy
Audio-Visual Center
Indiana University
Bloomington, IN 47405

Dr. E. James Hartzell
Army/Nasa Aircrew-Aircraft
Integration Program MS 239-9
NASA-Ames Research Center
Moffett Field, CA 94035

Dr. Bonnie E. John
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. William B. Johnson
Search Technology, Inc.
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092

Dr. Robin Kinkead
American Institutes for Research
45 North Rd.
Bedford, MA 01730

Prof. David L. Kleinman
Electrical Engrg & Systems Engrg Department
University of Connecticut
Storrs, CT 06268

Dr. Kathy Lang
Department of Psychology
Memphis State University
Memphis, TN 38152

Dr. Eugene F. Lynch
Principal Scientist, Tektronix Inc.
P.O. Box 500, M.S. 50-320
Beaverton, OR 97077

Dr. Jean McKendree
MCC Human Interface Laboratory
3500 W. Balcones Center Dr.
Austin, TX 78759

Mr. L. Hardy Mason
Usability Lab Administer
151 Farmington Avenue
Hartford, Connecticut 06156

Dr. Catherine R. Marshall
Director, Advanced User Interfaces
US West Advanced Technologies
6200 S. Quebec St.
Englewood, CO 80111

Ms. Joy Mountford
Apple Computer
20525 Mariani Ave, MS: 27A0
Cupertino, CA 95014

Dr. William H. Muto, Manager
Corporate Human Factors, R&D
Texas Instruments, Inc.
MS 8223
P. O. Box 655474
Dallas, TX 75265

Dr. Allen Newell
Department of Computer Science
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Mr. Jim Palmer
Apple Computer, Inc.
10500 No. DeAnza Blvd. MS: 27AR
Cupertino, CA 95014

Dr. Richard Pew
BBN Laboratories, Inc.
Experimental Psychol Dept.
70 Fawcett Street
Cambridge, MA 02238

Prof. Peter Polson
University of Colorado
Campus Box 345
Boulder, CO 80309-0345

Prof. Scott Robertson
Department of Psychology
Rutgers University
Busch Campus
New Brunswick, NJ 08903

Dr. William B. Rouse
Search Technology, Inc.
4725 Peachtree Corners Circle #200
Norcross, GA 30092

Professor Penelope M. Sanderson
1206 W. Green Street
Department of Mechanical and
Industrial Engineering
Urbana, IL 61801

Dr. H. Wallace Sinaiko
Manpower Research Advisory Services
Smithsonian Institution
801 N. Pitt Street, Suite 120
Alexandria, VA 22314-1713

Dr. Tamara L. Sturak
CFC, 271 Evans Hall
University of California
Berkeley, CA 94720

Dr. Martin A. Tolcott
Decision Science Consortium
1895 Preston White Drive
Suite 300
Reston, VA 22091-4369

Dr. Douglas Towne
U. of Southern California
Behavioral Technology Lab
1845 South Elena Avenue
Fourth Floor
Redondo Beach, CA 90277

Prof. Christopher D. Wickens
Department of Psychology
University of Illinois
Urbana, IL 61801

Dr. Wayne Zachary
CHI Systems, Inc.
Gynedd Plaza III Bethlehem Pike
Spring House, PA 19477